



Raspberry Pi (5) - gestion des INPUT -

Gestion des INPUT 1 : bouton PUSH

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

Objectif :	durée : 4h
Utilité :	

Matériel : plaque labdec – composants électroniques

Prérequis : Connexion à distance avec SSH, commande GPIO

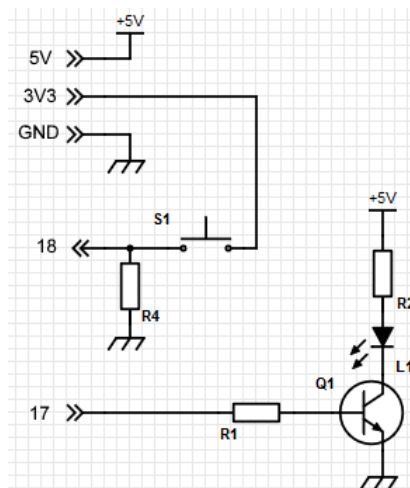
Compétences et savoirs principalement visées :
C2-1, C2-2 (page 3a), C3-2, C3-3 (page 3b à 6)

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

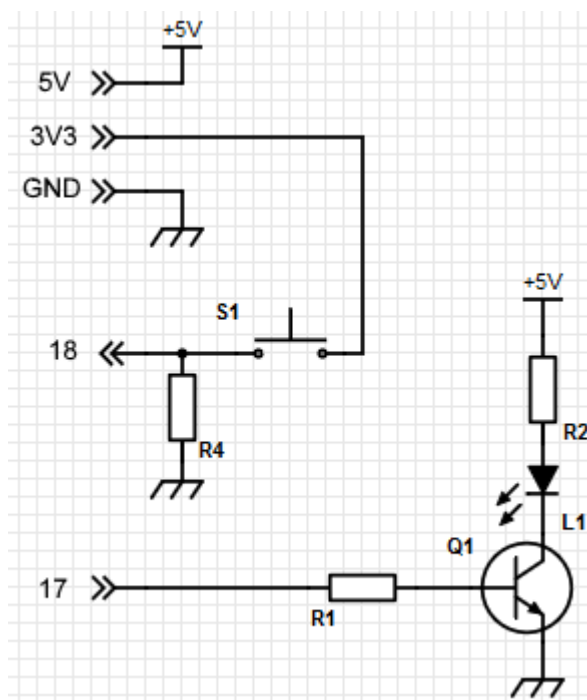


- Réalise le montage suivant sur la plaque Labdec

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** !!!

Le GPIO en mode INPUT

Tu connais maintenant les GPIO en mode **OUTPUT**, qui permet d'allumer ou éteindre un équipement électrique.

Le mode **INPUT**, quant à lui, permet **d'entrer des informations dans le RPi** ; par exemple : l'état d'un bouton poussoir (appuyé/relâché), l'état d'un capteur quelconque (actif/inactif), ou simplement des bits provenant d'un autre équipement électronique via un câble.

Sur ta plaque LabDec, tu as câblé un bouton-poussoir (BP) relié au signal 18 (bcm) 

- ◆ Tapes la commande suivante et indiques ce que tu constates :

Note importante : tout ce qui est situé sur la ligne après le caractère « # » est un commentaire ; il ne faut pas le taper

```
gpio readall # Vérifier l'état des broches
```

Dans quel état est la broche 18: (IN ou OUT ?) _____, (0 ou 1 ?) _____

- ◆ La broche 18 est-elle bien en mode INPUT ? Sinon quelle commande aurait-tu tapé pour la passer en mode INPUT ?

- ◆ Tapes la commande suivante, **tout en appuyant sur le BP** et indiques ce que tu constates :

```
gpio readall # Vérifier l'état des broches
```

Dans quel état est la broche 18: (IN ou OUT ?) _____, (0 ou 1 ?) _____

Pour comprendre ce qui se passe au niveau électrique, rempli le tableau suivant en faisant les mesures correspondantes :

Bouton S1	Tension au point 18 <i>en Volts *</i>	Niveau logique signal 18 #
relâché		
appuyé		

* à mesurer

à déduire

Détection de l'appui et du relachement depuis un script

La commande pour lire l'état (0 ou 1) d'un GPIO en mode input est :

```
gpio -g read n°du signal
```

- ◆ Tapes la commande suivante, **tout en laissant le BP relâché**, et indique quelle est la valeur renvoyée par la commande :

```
gpio -g read 18 # lecture de l'état de l'entrée 18
```

quelle est la valeur renvoyée par la commande : _____

- ◆ Tapes la commande suivante, **tout en appuyant sur le BP** et indique quelle est la valeur renvoyée par la commande :

```
gpio -g read 18 # lecture de l'état de l'entrée 18
```

quelle est la valeur renvoyée par la commande : _____

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

Fichier read.sh 

```
#!/bin/bash
gpio -g mode 18 in
b=0
while [ 1==1 ]
do
# attendre que le BP soit poussé
while [ $b -eq 0 ]
do
b=$(gpio -g read 18)
done
echo "le bouton est appuyé..."

# attendre que le BP soit relaché
while [ $b -eq 1 ]
do
b=$(gpio -g read 18)
done
echo "le bouton est relaché..."

done
```

Condition de sortie : $\$b \neq 0$

$\$b$ va prendre la valeur 1 quand BP sera appuyé

La boucle s'arrête quand $\$b=1$, c'est-à-dire quand BP est appuyé

- ◆ Rendre «**read.sh**» exécutable, puis teste-le.

Bouton PUSH

l'appui sur le bouton PUSH provoque l'allumage de la led 17 ; elle s'éteint dès qu'on relâche le bouton

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

```
Fichier push.sh
#!/bin/bash

gpio -g mode 17 out
gpio -g mode 18 in
b=1

while [ 1==1 ]
do
  b=$(gpio -g read 18)
  gpio -g write 17 $b
  sleep 0.2
done
```

L'instruction **while** => **do** => **done** permet de faire une boucle de programme ; une boucle est une série d'instructions qui va s'exécuter en boucle jusqu'à ce que la condition de sortie soit remplie ;

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

```
cd /root
chmod +x push.sh
```

- ◆ Lances ton script avec la commande : **./ push.sh**
- ◆ Que constate-tu ? (Relance le script autant de fois que nécessaire)
- ◆ Rempli le tableau pour vérifier que tu as compris le fonctionnement :

Bouton S1	Niveau logique broche 18	État de la LED <i>allumée/éteinte</i>	
relâché			
appuyé			

- ◆ A ton avis, est-il nécessaire d'utiliser un Raspberry Pi pour obtenir ce fonctionnement ?

- ◆ Si le script est arrêté, la LED s'allume-t-elle encore quand on appui sur le bouton ? Pourquoi ?