



# Raspberry Pi (3) - Script BASH -

## SCRIPT BASH DE PROGRAMMATION DES GPIO

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

<b>Objectif :</b> Apprendre à contrôler les GPIO avec un script BASH	<b>durée :</b> 4h
<b>Utilité :</b> Les GPIO permettent la commande de composants électriques tels que des relais 220V, que ce soit via une commande manuelle (bouton poussoir) ou automatique (ex : programmation horaire) ; cette commande doit impérativement se faire par un <b>script</b> (langage interprété) ou avec un <b>logiciel</b> (langage compilé)	

**Matériel :** alimentation de laboratoire – multimètre – plaque labdec – composants électroniques  
**Prérequis :** Connexion à distance avec SSH

**Compétences et savoirs principalement visés :**

**Travail à réaliser :**  
 - temporisation, phénomène cyclique, fréquence, rémanence, rapport cyclique

**Schéma du système :**

Pi B+ GPIO Ref

1	3.3V	2	5V
2		3	5V
3		4	GND
4		5	GND
5		6	GND
6		7	GND
7		8	GND
8		9	GND
9		10	3.3V
10		11	GND
11		12	GND
12		13	GND
13		14	GND
14		15	GND
15		16	GND
16		17	GND
17		18	GND
18		19	GND
19		20	GND
20		21	GND

$R1=R3=47k\Omega$   
 $R2=R4=470\Omega$

# LED CLIGNOTANTE

## SCRIPT ? Qu'est-ce c'est ?

Jusqu'à présent, tu as commandé les GPIO en ligne de commande, depuis un terminal Linux ; dans la vie de tout les jours, ce ne serait **pas très pratique** !! de plus les possibilités d'utilisation sont limitées.

Un **script** est un petit programme qui permet **d'automatiser une série de commande** ; il simplifie donc l'utilisation des GPIO, mais permet aussi de faire des choses plus complexes (ex : des temporisations précises)

Grâce à la programmation de scripts, tu vas pouvoir faire plus chose avec les GPIO, et l'utilisation sera plus simple

**Retenir** : dans la pratique, les GPIO seront toujours commandés par un script ou un logiciel

## Conseils :

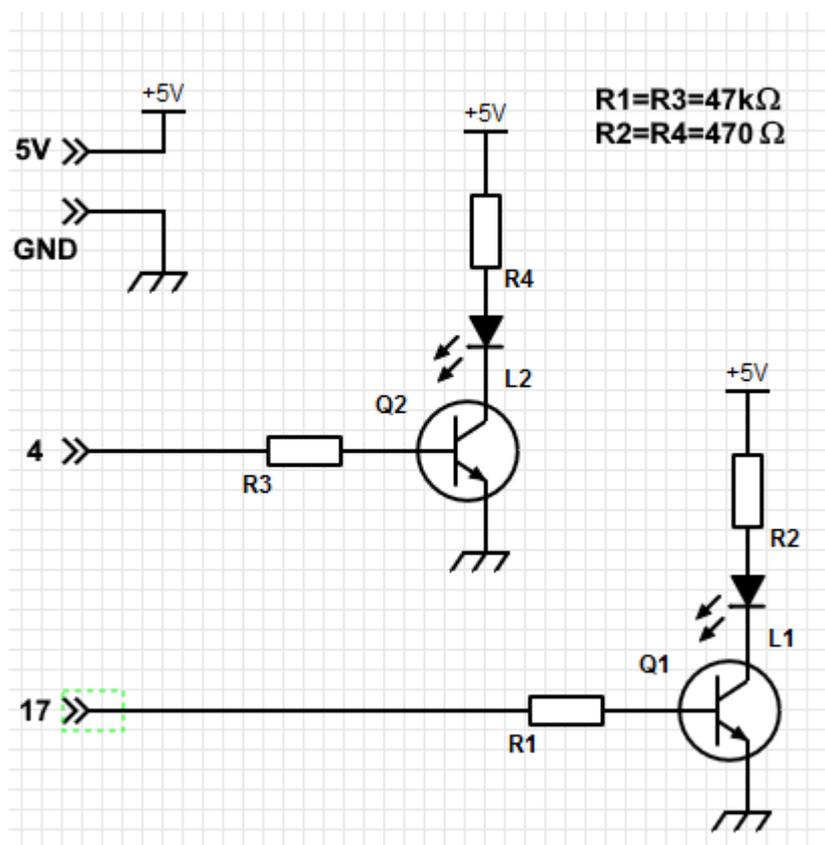
1. si quelque chose ne fonctionne pas, ne t'affole pas : revérifie depuis le début si tu n'a pas oublié un détail.

- ◆ 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



## Une LED clignotante, comment ça marche ?

Allumer ou éteindre une LED connectée à un GPIO en ligne de commande, tu sais le faire... et maintenant tu vas apprendre à la faire clignoter à une fréquence que tu auras définie

Une LED clignotante, qu'est-ce que c'est ? C'est une LED qui s'allume et s'éteint de manière **cyclique\***, à une **fréquence** \* fixée.

Le temps « allumé » et le temps « éteint » peuvent être égaux ou non

\* donne la définition de ces termes (**cyclique**, **fréquence**) :

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

Fichier clignote.sh

```
#!/bin/bash

gpio -g mode 4 out
gpio -g mode 17 out
for (( i=1; i<=5; i++ ))
do
  gpio -g write 4 1
  gpio -g write 17 1
  sleep 0.5
  gpio -g write 4 0
  gpio -g write 17 0
  sleep 0.5
done
```

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

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

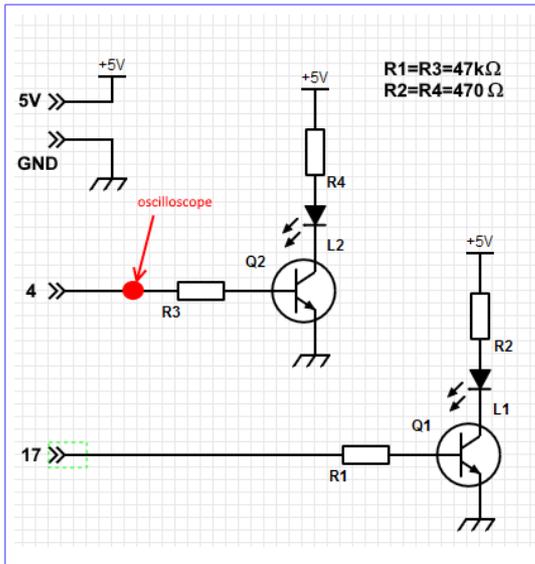
- ◆ Lances ton script avec la commande :

```
./clignote.sh
```

- ◆ Que constate-tu ? (Relance le script autant de fois que nécessaire)

Combien de fois les LEDs clignent-elles ?

◆ Visualisation du signal à l'oscilloscope

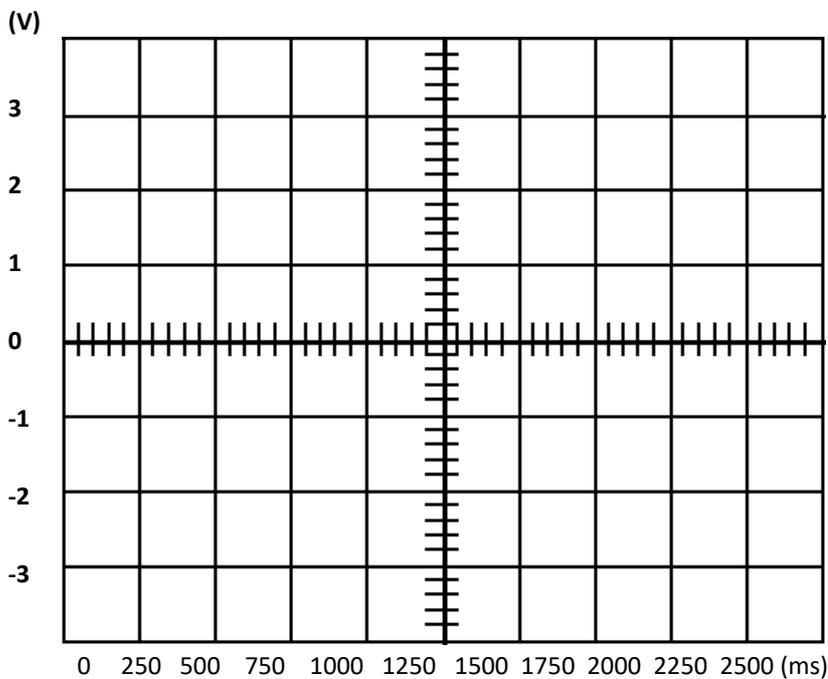


Pour comprendre ce qui se passe au niveau de la led L2, tu vas regarder **le signal électrique présent au point 4**. Ce signal commande le transistor qui commande L2

La sonde de l'oscilloscope est placée comme indiqué sur le schéma

Puis tu lances le script, et quand le signal s'affiche, appuie sur le bouton « mémoire »

Reproduit le signal observé à l'oscilloscope sur le dessin ci-dessous ;



Réglages :

Vertical : 1V/div

Horizontal : 250ms/div

En regardant cet oscillogramme, explique pourquoi la led L2 clignote.

Sur l'oscillogramme ci-dessus, surligne en rouge un cycle complet.

- ◆ La LED clignote donc 5 fois, puis s'arrête ; Regardes le **code source à la ligne 5** ; modifie légèrement cette ligne pour que la LED clignote **10 fois** :

Recopie ici la ligne modifiée

- ◆ Regardes le **code source à la ligne 9** ; **combien de temps les LEDs restent-elles à l'état allumé ?** (*pour comprendre le rôle de la commande « sleep », fais une recherche Google avec « man sleep »*)

- ◆ Modifie légèrement la ligne 9 pour que les LEDs restent à l'**état allumé pendant 1 seconde** ; testes ce script modifié ; que constates-tu ?

Recopie ici la ligne modifiée

- ◆ Le problème, c'est que maintenant l'état allumé est plus long que l'état éteint ; repère la ligne qui règle la durée de l'état éteint ( cette durée est de 0,5 seconde) ; quel est le numéro de cette ligne ?

- ◆ Modifie légèrement cette ligne pour que les LEDs restent à l'**état éteint pendant 1 seconde** ; testes ce script modifié ; que constates-tu ?

Recopie ici la ligne modifiée

- ◆ **Exercice 1** : on souhaite que la LED clignote très vite ; le problème, c'est que si le clignotement est trop rapide, l'œil humain voit la LED « allumée » en permanence (phénomène de **rémanence**); modifie le script de façon à ce que les LED clignent *le plus vite possible en évitant le problème de rémanence* ; pour cela tu vas pouvoir tester plusieurs valeurs et remplir le tableau ci-dessous ; quand le tableau est rempli, entoure en rouge la valeur minimale acceptable.

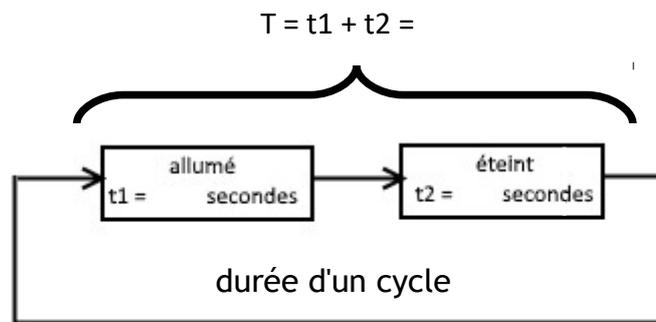
Durée allumage (s)	0,2	0,1	0,05	0,03	0,01
acceptable ou non acceptable					

- ♦ **Exercice 2** : Le phénomène de rémanence est au contraire utile dans le domaine de la vidéo pour déterminer le nombre minimal d'images à afficher par seconde (on ne doit pas voir l'image clignoter) ; au **cinéma**, il est de 30 images/s (30 Hz), en **télévision** il est de 25 images/s (25 Hz), 50 images/s (50 Hz) ou 100 images/s (100 Hz).

As-tu remarqué que la fréquence (en Hertz) correspond à un nombre de **cycles par seconde** ?



Pour la LED de l'exercice précédent, la **fréquence** de clignotement correspond donc au **nombre de clignotements par seconde** ; pour la calculer, on doit d'abord **calculer la durée T d'un seul clignotement** ( complète le schéma en utilisant les valeurs entourée en rouge dans l'exercice 1 ) :



Une fois qu'on connaît la durée du cycle, on obtient la fréquence de clignotement avec la formule :

$$f = 1/T$$

Calcule la fréquence de clignotement de la LED :

$$f = \quad \text{Hz}$$

Exercice 3 : soit une LED clignotante avec le cycle suivant :

## FLASH LUMINEUX

### Une flash lumineuse, qu'est-ce c'est ?

Dans l'exercice précédent, tu as fait clignoter une LED à l'aide d'un script ; dans ce script, on avait défini une valeur identique pour la durée « allumé » et la durée « éteint »...

Maintenant, si tu modifies le script en définissant une valeur plus petite pour la durée « allumée », tu obtiens un flash lumineux ; les flashes sont utilisés car ils consomment moins de courant et attirent davantage l'attention.

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

Fichier clignote.sh

```
#!/bin/bash

gpio -g mode 4 out
gpio -g mode 17 out
for (( i=1; i<=5; i++ ))
do
  gpio -g write 4 1
  gpio -g write 17 1
  sleep 0.3
  gpio -g write 4 0
  gpio -g write 17 0
  sleep 1.2
done
```

- ◆ Testes le script ; qu'est-ce que cela donne ? comment qualifierais-tu cette lumière ? Quel est l'intérêt de ce mode de fonctionnement quand le montage est alimenté par batterie ?

## Rapport cyclique

On appelle rapport cyclique la valeur :

$$n = \frac{t_h}{T}$$

où  $t_h$  est la durée à l'état haut et  $T$  la durée totale du cycle

**Exercice** : calcule le rapport cyclique d'un éclairage clignotant qui est allumé pendant 1 seconde, puis éteint pendant 1 seconde.

## Récapitulatif

Tu sais maintenant comment programmer une sortie sur le Raspberry Pi

## Objectif suivant

Tu vas maintenant te perfectionner dans ta connaissance des GPIO en mode INPUT (entrée)

**Bonus** : modifie le programme de façon à ce que les deux leds s'allument alternativement (c'est à dire L2 est éteinte quand L1 est allumée et réciproquement) ; le rapport cyclique d'éclairage doit être de 0,5 ; le nombre de cycle est fixé à 100

Recopie ici les lignes modifiées :