UFR SCIENCES ET TECHNOLOGIES

INFORMATIQUE

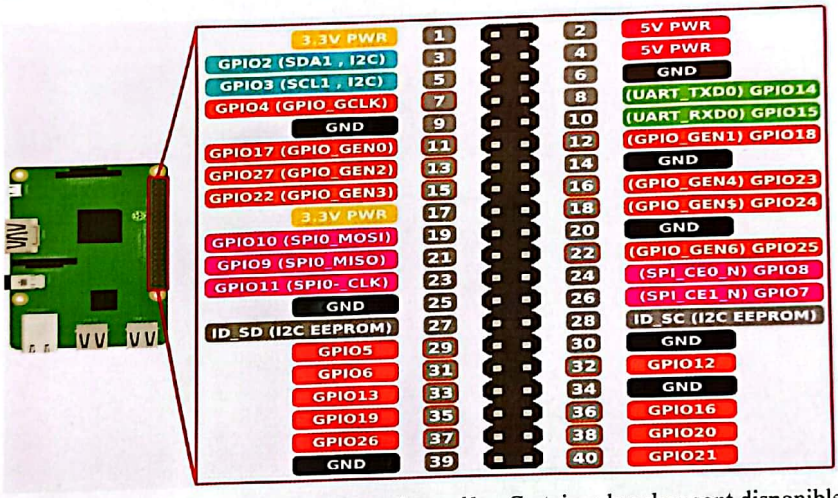
Année universitaire 2022 / 2023

L2I LICENCE 3
TP Internet des Objets
Fiche de TP N° 1
Projet 1 : Circuit électrique – Clignoter une LED avec Raspberry Pi 4

Maîtriser le codage, ce n'est pas uniquement réaliser des projets à l'écran : vous pourrez également contrôler tous les composants électroniques rattachés aux broches du port GPIO (General Purpose Input/Output) d'un Raspberry Pi qui est un excellent outil pour apprendre l'informatique physique.

Présentation du port GPIO

Situé sur le bord supérieur de la carte du Raspberry Pi, ou à l'arrière dans le cas du Raspberry Pi 400, et consistant en deux longues rangées de broches métalliques, le port GPIO (general-purpose input/output) vous permet de connecter des éléments matériels comme des LED et des interrupteurs au Raspberry Pi pour commander les programmes que vous créez. ***Les broches peuvent être utilisées aussi bien en entrée qu'en sortie.***



Le port (ou connecteur) GPIO est constitué de ***40 broches mâles***. Certaines broches sont disponibles pour vos projets d'informatique physique, d'autres sont consacrées à l'alimentation(Anode(3.3V ou 5V), Cathode (GND=Ground=Masse), d'autres encore sont réservées pour communiquer à l'aide d'éléments complémentaires comme la Sense HAT (Carte complémentaire multifonctionnelle pour Raspberry Pi, équipée de capteurs et d'un écran matriciel à LED).

Il existe plusieurs catégories de types de broches, chacune ayant une fonction particulière :

Type de broche	Catégorie	Fonction
3V3	Alimentation 3,3 volts	Une source d'alimentation permanente de 3,3 V, la même tension alimentant votre Raspberry Pi en interne
5V	Alimentation 5 volts	Une source d'alimentation permanente de 5 V, la même tension alimentant votre Raspberry Pi via le connecteur microUSB
Masse(GND)	Masse 0 volts	Une prise de masse, utilisée pour compléter un circuit relié à une source d'énergie
GPIO XX	Numéro de broche d'entrée/sortie à usage général « XX »	Les broches GPIO disponibles pour vos programmes, identifiées par un numéro compris entre 2 et 27
ID EEPROM	Broches réservées à des fins particulières	Broches réservées à Hardware Attached on Top (HAT) et autres accessoires

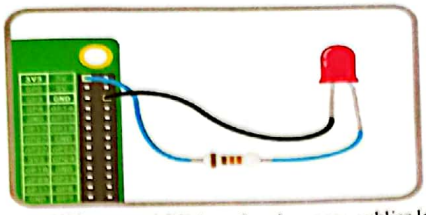
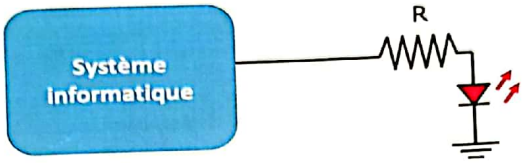


Figure TP1-1 : Câblez votre LED à ces broches, sans oublier la résistance

Si votre Raspberry Pi est sous tension, la LED devrait s'allumer. Si elle ne s'allume pas, vérifiez votre circuit : assurez-vous que vous n'avez pas utilisé une valeur de résistance trop élevée, que tous les fils sont correctement connectés et que vous avez bien choisi les bonnes broches GPIO telles qu'elles sont illustrées sur le schéma.

b. Allumage et extinction d'une LED via un système informatique

Pour réaliser ce fonctionnement à l'aide d'un système informatique, il convient d'utiliser un dispositif d'entrée/sortie (E/S).



Le système informatique pilote l'allumage et l'extinction de la LED par application de deux niveaux de tension électrique

Une fois que votre LED fonctionne, il est temps de la programmer. Débranchez le fil de raccordement de la broche 3,3 V et connectez-le à la broche GPIO 25 (étiquetée GP25 sur la Figure TP1-2). La LED s'éteint, ce qui est parfaitement normal, rassurez-vous.

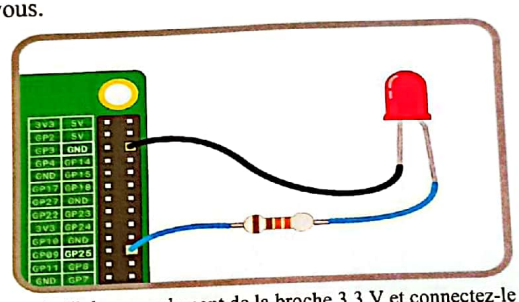


Figure TP1-2 : Débranchez le fil de raccordement de la broche 3,3 V et connectez-le à la broche 25 du GPIO

Vous êtes maintenant prêt à créer un programme Scratch ou Python pour allumer et éteindre votre LED.

LedClignotante.py

```
# importer la section LED de la bibliothèque GPIO ZERO pour utiliser les broches du GPIO sous Python
from gpiozero import LED
from time import sleep

#Indiquons à GPIO ZERO la broche à utiliser; ici 25
led = LED(25)

#Commandons les LED

while(True) :
    led.on() # Pour allumer les LED
    sleep(1) # Attendre 1 seconde
    led.off() #Pour eteindre la LED
    sleep(1)
```

Références

Brancher une DEL au Raspberry Pi : https://apical.xyz/fiches/le_gpio_general_purpose_input_output/brancher_une_del_au_raspberry_pi

Alimentation d'une LED : http://www.zpag.net/Electroniques/Diode/AlimDEL.htm#:~:text=La%20tension%20aux%20bornes%20de,allume%20bien%20comme%20il%20faut.

Alimentation d'une LED : http://www.zpag.net/Electroniques/Diode/led.htm

C. Un interrupteur

Un ***interrupteur à bouton-poussoir***, ou ***interrupteur instantané***, est le type d'interrupteur utilisé pour commander une console de jeux. Disponible en deux ou quatre bornes (les deux types fonctionnent avec le Raspberry Pi), le bouton-poussoir est un dispositif d'entrée, qui vous permet de transmettre des ordres pour que votre dispositif effectue une tâche.

Un autre type d'interrupteur est le commutateur : alors qu'un bouton poussoir n'est actif que lorsque vous le maintenez enfoncé, un commutateur (par exemple un interrupteur d'éclairage) s'active en l'appuyant une fois, puis reste actif jusqu'à ce que vous l'appuyiez à nouveau.

D. Une diode électroluminescente(LED)

K = (K)Cathode, pôle "négatif" de la LED, patte la plus courte.

A = Anode, pôle "positif" de la LED, patte la plus longue

Une ***LED (Light Emitting Diode)*** ou ***DEL (Diode Electroluminescente)*** est un périphérique de sortie que vous contrôlez directement à partir de votre programme. La LED est un composant dit passif, de la famille des semi-conducteurs (comme la diode). Il s'agit d'une diode un peu particulière, qui a la propriété d'émettre de la lumière quand un courant la parcourt (de l'Anode vers la Cathode). Vous en trouverez partout dans votre maison, des petites qui vous indiquent que vous avez laissé votre machine à laver allumée, aux grandes qui éclairent vos chambres.

Les LED sont disponibles dans une large gamme de formes, de couleurs et de tailles, mais toutes ne sont pas compatibles avec le Raspberry Pi : *évitez les LED conçues pour une alimentation de 5 ou 12 V*.

E. Les résistances

Les ***résistances*** sont des éléments qui contrôlent le flux de courant électrique et sont disponibles en différentes valeurs mesurées à l'aide d'une unité de mesure appelée ohm(Ω). ***Plus le nombre d'ohms est élevé, plus la résistance est importante.*** Pour les projets d'informatique physique Raspberry Pi, elles sont surtout utiles à protéger les LED contre une consommation de courant trop importante et contre les dommages qu'elles pourraient causer à elles-mêmes ou à votre Raspberry Pi. Même si la DEL supporte la tension fournie par le Pi sur la broche de 3.3V, il faut utiliser une résistance (d'environ 330 Ω) pour protéger la DEL et le Raspberry Pi. En effet, une fois la DEL allumée, elle risque de consommer trop de courant (le I dans V=RI) et de finir par faire griller soit la DEL, soit le Pi.

Pour connaître la valeur requise pour la résistance soit :

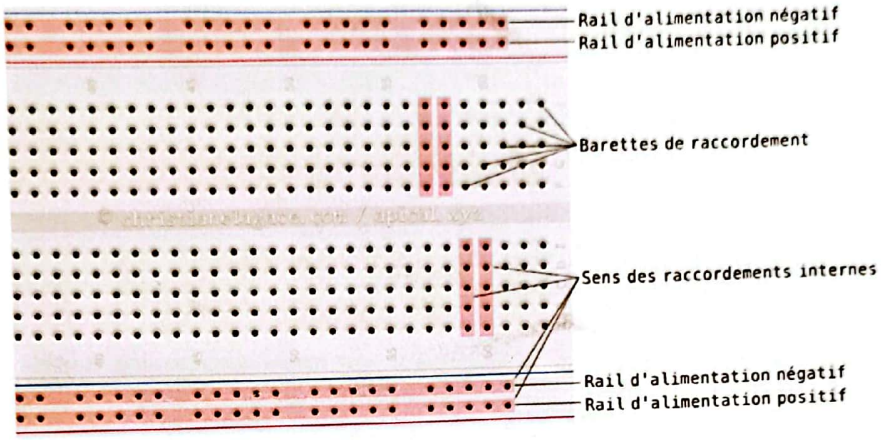
- consultez la ***fiche technique de la DEL*** afin de connaître la tension maximale qu'elle peut accepter.
- utiliser un ***petit outil en ligne*** pour calculer la valeur de la résistance à utiliser. En voici quelques-uns :
  - https://www.hobby-hour.com/electronics/ledcalc.php
  - http://www.muzique.com/schem/led.htm

  NB : Il faut utiliser une résistance égale ou plus grande que la valeur donnée dans le calcul. Mais si la valeur est trop élevée, il risque de ne pas y avoir assez de courant pour allumer la DEL.
- Maintenant, quand on a *une résistance en main, et qu'elle est détachée du petit papier qui donnait sa valeur*, comment peut-on trouver cette information?
- Il est possible de ***déterminer la valeur d'une résistance à partir de ses bandes de couleur***.

Composants électroniques

Le port GPIO n'est que l'une des parties dont vous aurez besoin pour explorer le domaine de l'informatique physique, l'autre partie étant constituée de ***composants électroniques***, les dispositifs que vous pourrez contrôler à partir du port GPIO. Il existe des milliers de composants, mais la plupart des projets GPIO sont réalisés à partir des parties communes ci-après.

A. Une planche de maquettage



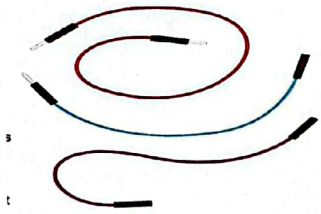
une planche de maquettage (aussi appelée platine d'essai, platine d'expérimentation, platine de prototypage ou, en anglais, breadboard) peut faciliter considérablement tous vos projets d'informatique physique. Plutôt que de travailler avec différents composants séparés qui doivent être reliés par des fils, une platine vous permet d'insérer des composants et de les connecter par des circuits métalliques cachés sous la surface. *Une platine n'est pas indispensable pour vous lancer dans l'informatique physique, mais elle vous sera certainement très utile.*

Les rails d'alimentation sont les deux premières et les deux dernières rangées de trous sur le côté long de la planche de maquettage. Le rail négatif est généralement marqué d'une ligne bleue ou noire.

La couleur des fils utilisés pour brancher les composants sur la planche de maquettage n'a pas d'importance. Cependant, par convention, on utilisera un fil noir pour brancher à la mise à terre (ground ou GND).

Sur la planche de maquettage, le fil de mise à terre sera généralement, mais pas obligatoirement, branché dans le rail d'alimentation négatif.

B. Les fils de raccordement



Les ***fils de raccordement***, ou ***cordons de raccordement***, servent à connecter les différents composants à votre Raspberry Pi et, si vous n'utilisez pas de platine, les uns aux autres. Il en existe trois types :

a. ***mâle-femelle (M2F)***, qui servent à connecter la platine aux broches GPIO ;
b. ***femelle-femelle (F2F)***, nécessaires à connecter des composants individuels entre eux si vous n'utilisez pas de platine ;
c. ***mâle-mâle (M2M)***, utilisés pour effectuer des connexions des différentes parties de la platine entre elles.

En fonction de votre projet, vous pourriez avoir besoin des trois types de fils de raccordement; *si vous utilisez une platine, des fils de raccordement M2F et M2M devraient suffire.*

BoutonPoussoir.py

```
from gpiozero import Button  # importer la section Button de la bibliothèque GPIO ZERO pour utiliser
                             #les broches du GPIO sous Python

from gpiozero import LED
from time     import sleep

button = Button(2) #Indiquons GPIO ZERO la broche à utiliser; ici 2

# GPIO Zero dispose de la fonction wait_for press pour que le code soit exécuté quand le bouton est appuyé

button.wait_for_press()
print("Vous avez appuyé !")
```

Cliquez sur le bouton Exécuter, puis appuyez sur le bouton-poussoir. Votre message s'affichera dans le shell de Python en bas de la fenêtre Thonny : vous avez réussi à lire un message transmis par une broche GPIO !

Pour étendre encore votre programme afin de contrôler une LED et via un bouton, ajoutez la LED et la résistance dans le circuit si vous ne l'avez pas encore fait : n'oubliez pas de connecter la résistance à la broche GPIO 25 et à la patte longue de la LED, et la patte plus courte de la LED au rail de masse de votre platine.
Retournez au début de votre programme et modifier le pour avoir ce qui suit :

BoutonPoussoir.py

```
from gpiozero import Button  # importer la section Button de la bibliothèque GPIO ZERO pour utiliser les broches du GPIO so

from gpiozero import LED
from time     import sleep

button = Button(2) #Indiquons GPIO ZERO la broche à utiliser; ici 2

# GPIO Zero dispose de la fonction wait_for press pour que le code soit exécuté quand le bouton est appuyé

led = LED(25)

while True :
    button.wait_for_press()
    led.on()
    sleep(3)
    led.off()
```

Cliquez sur le bouton Exécuter, puis appuyez sur le bouton-poussoir : la LED s'allume pendant trois secondes, puis s'éteint et le programme s'arrête. Félicitations : vous pouvez contrôler une LED en utilisant une entrée de bouton dans Python !

Références

Périphériques d'entrée : https://gpiozero.readthedocs.io/en/stable/api_input.html

Parmi les autres composants électriques courants, citons les *moteurs*, qui nécessitent une carte de contrôle spéciale avant de pouvoir être connectés au *Raspberry Pi*, les *capteurs infrarouges* qui détectent les mouvements, les *capteurs de température et d'humidité* qui peuvent être utilisés pour prévoir le temps, et les *photorésistances (LDR)*, des *dispositifs d'entrée* qui fonctionnent comme des LED inversées en détectant la lumière.

Pour réaliser les projets de ce chapitre, vous devez au minimum posséder :

- 3 LED : rouge, verte, et jaune ou ambrée
- 2 interrupteurs à bouton-poussoir
- 1 buzzer actif
- des fils de raccordement mâle-femelle (M2F) et femelle-femelle (F2F)
- en option, une platine et des fils de connexion mâle-mâle (M2M)

TP4 Feux de circulation

Maintenant que vous savez comment utiliser les boutons, les buzzers et les LED en entrée et en sortie, vous êtes prêt pour vous lancer dans l'informatique du monde réel : les feux de circulation et le bouton sur lequel vous pouvez appuyer pour traverser la route. Pour ce projet, vous aurez besoin d'une platine ; une LED rouge, une verte et une orange; trois résistances 330 Ω ; un buzzer ; un interrupteur à bouton-poussoir ; plusieurs câbles de raccordement mâle-mâle (M2M) et mâle-femelle (M2F).

Commencez par construire le circuit (Figure TP4-1), reliant le buzzer à la broche GPIO 15 (GP15 dans la Figure TP4-1), la LED rouge à la broche GPIO 25 (GP25), la LED jaune à la GPIO 8 (GP8), la LED verte à la GPIO 7 (GP7) et l'interrupteur à la GPIO 2 (GP2). N'oubliez pas de connecter les résistances 330 Ω entre les broches GPIO et les pattes longues des LED, et de relier les secondes pattes de tous vos composants au rail de masse de votre platine. Pour finir, connectez le rail de masse à une broche de masse (étiquetée GND) sur le Raspberry Pi pour compléter le circuit.

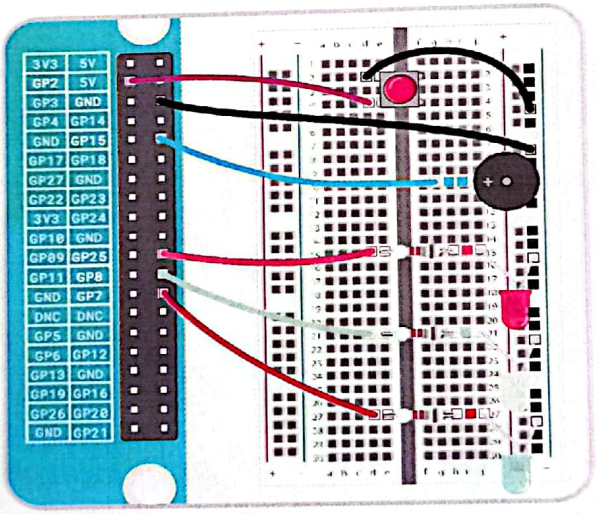


Figure TP4-1 : Schéma de câblage pour le projet Feux de circulation

a. Projet Python : Feux de circulation

Créez un nouveau projet dans l'IDE Python et enregistrez-le sous Feux Circulation, puis saisissez le code suivant :

FeuxCirculation.py

```
#Exécutez le programme, et observez vos LED : d'abord le rouge s'allume,
#puis le rouge et l'orange, puis le vert, puis l'orange,
#et pour finir la séquence recommence avec le feu rouge.
#Cette séquence correspond à celle utilisée par les feux de circulation au Royaume-Uni;
#si vous le souhaitez, vous pouvez modifier la séquence pour l'adapter à celle d'autres pays.

from gpiozero import LED
from gpiozero import Button
from gpiozero import Buzzer
from time     import sleep

#Définir les numéros des ports de sortie des LED
ledRouge = LED(25)
ledJaune = LED(8)
ledVerte = LED(7)

#Définir le numéro de port de sortie du Buzzer

buzzer = Buzzer(15)

#Définir le numéro de port d'entrée du bouton poussoir
bouton = Button(2)

#Définissons les séquences de feux de circulation
while True:
    ledRouge.on()
    sleep(5)
    ledJaune.on()
    sleep(2)
    ledRouge.off()
    ledJaune.off()
    ledVerte.on()
    sleep(5)
    ledVerte.off()
    ledJaune.on()
    sleep(5)
    ledJaune.off()
```

Cliquez sur le bouton Exécuter et votre système de feux de circulation tricolore se mettra à réguler la circulation.

b. Projet Python : Feux de circulation avec Passage piéton

A faire à la maison : En utilisant le schéma de câblage pour le projet Feux de circulation (Figure TP4-1), complétez le code pour simuler un passage piéton. Vous avez besoin de votre programme qui surveille à quel moment le bouton est appuyé

Références

Contrôler une séquence de feux de circulation avec Python : https://projects.raspberrypi.org/fr-CA/projects/traffic-lights-python/4