

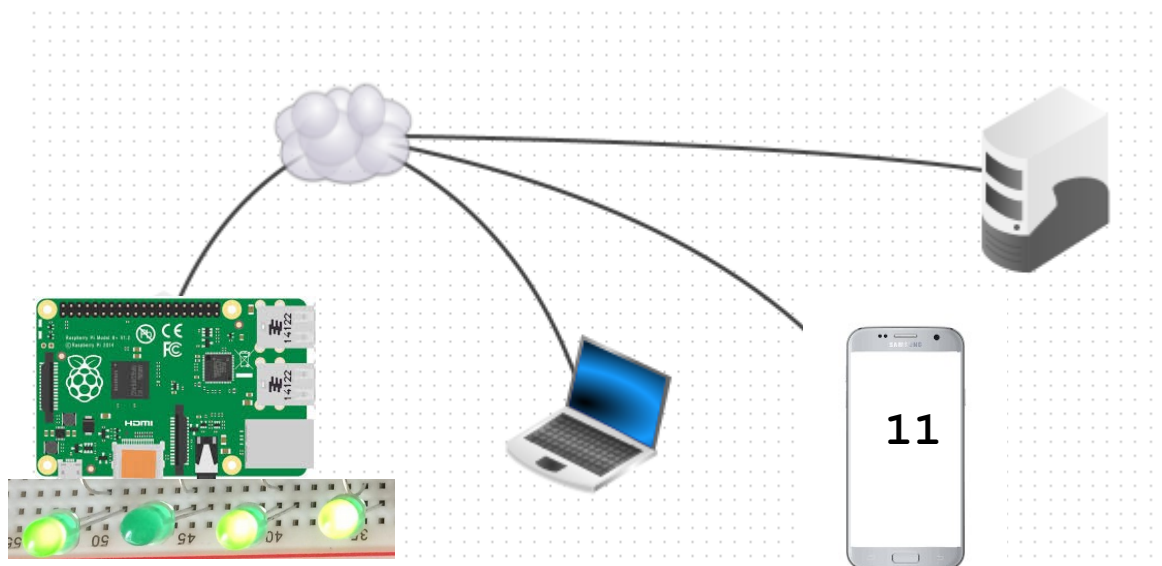
Comptage en binaire

Fiche élève

Objectif

Cette séance a pour objectif de réaliser un montage composé de LEDs dont l'allumage sera déclenché au travers d'un script Python exécuté sur un Raspberry Pi relié au montage. L'allumage devra correspondre à la numération binaire (sur 4 bits dans ce TP) d'un nombre entier qui sera passé en paramètre.

Dans la deuxième partie de la séance, il est demandé d'utiliser le réseau local afin de mettre en place un environnement client-serveur : le Raspberry Pi fera office de serveur et les autres équipements (les clients) du réseau devront être capables de déclencher l'allumage des LEDs correspondant à un nombre entier qui sera transmis par le client.



Organisation

Le travail se fera en binôme.

Chaque binôme dispose d'un ordinateur de la salle ainsi que d'un Raspberry Pi.

Une platine de prototypage et quelques composants (LEDs, résistances et câbles de connexion) sont à disposition.

À la fin de chaque étape, faites le point avec le professeur avant d'avancer.

I. Préliminaire

Pour cette séance, vous deviez implémenter la fonction suivante :

```
def conversion(entier):  
    """ entier est un nombre entier positif.  
    Retourne l'écriture de entier en binaire sur un demi octet.  
    """
```

II. Allumage des LEDs

1. Montage

Effectuer votre montage de 4 LEDs sur une platine de prototypage (pensez à l'utilisation des résistances).

2. Le port GPIO

Reliez vos LEDs aux broches du Raspberry Pi qui sera utilisé, dans cette partie, comme un micro-contrôleur.

Vous pourrez effectuer une recherche afin de comprendre le fonctionnement du port d'entrées-sorties (ou GPIO) du Raspberry Pi.

3. Programmation

En utilisant votre IDE Python sur le Raspberry Pi, adaptez votre script contenant la fonction `conversion()` afin de pouvoir, lors de l'appel de votre fonction, déclencher l'allumage de vos LEDs et obtenir une représentation binaire « visuelle » de votre nombre entier passé en paramètre.

L'utilisation de la bibliothèque `RPi.GPIO`, installé sur votre Raspberry Pi, vous sera probablement d'une grande utilité.

Les liens suivants donnent quelques indications pour l'utilisation de la bibliothèque :

- <https://sourceforge.net/p/raspberry-gpio-python/wiki/Home/>
- <https://deussyss.developpez.com/tutoriels/RaspberryPi/PythonEtLeGpio/>

III. Client-serveur

Il faut maintenant modifier votre script afin que celui-ci puisse jouer le rôle de serveur et répondre à des requêtes envoyées par les équipements du réseau.

Les liens suivants donnent quelques indications pour commencer à utiliser la bibliothèque `flask` qui permet de construire des applications web simples :

- <https://flask.palletsprojects.com/>
- <http://sdz.tdct.org/sdz/creez-os-applications-web-avec-flask.html>

Vous pouvez commencer à tester votre script localement. Vous pouvez tester votre script initial de `conversion` (ne faisant pas référence au déclenchement de LEDs) sur votre ordinateur.

Une fois que vous avez compris le fonctionnement de la bibliothèque `flask`, il ne vous reste plus qu'à écrire votre script final permettant de lancer votre serveur sur le Raspberry et envoyer vos requêtes depuis les équipements du réseau local.