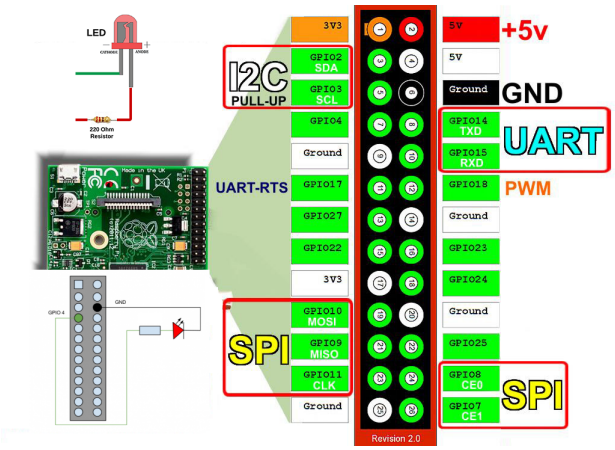
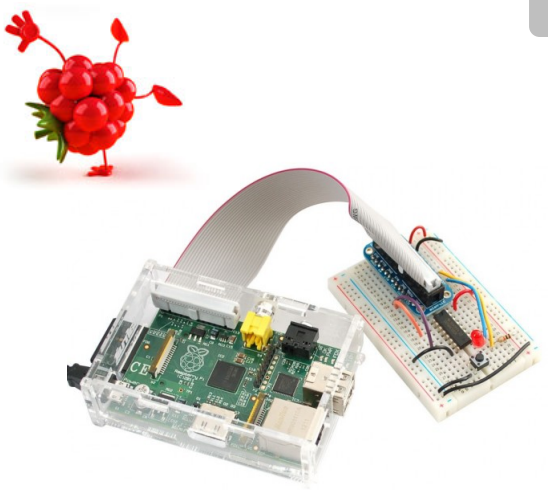
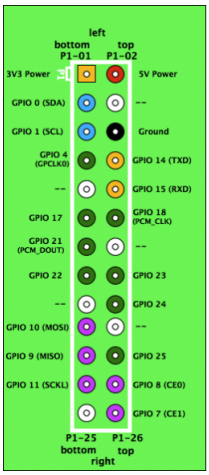





# Objectifs

⇝ Utiliser les entrées/sorties (GPIO) du Raspberry Pi pour interfacer du matériel extérieur.

⇝ Créer un serveur WEB affichant deux boutons ON et OFF sur lesquels l'utilisateur peut agir pour commander une LED en utilisant un PI-Cobbler [1] et Python.
Assurez-vous d'avoir mis votre environnement Python à jour et **la librairie** RPi.GPIO [2] qui permet d'utiliser les `E/S` du RPi assez facilement.

# Organisation

✓ **En binôme** : un « élève ressource » sur un ordinateur de la salle, associé à un « élève utilisateur » du Raspberry Pi ; (*En attendant la « mise à jour » de la salle ISN*).

**Conclusion** du TP_Python_raspberry_Pi : le Raspberry Pi est plutôt un **système embarqué**, l'interface console directe via le réseau (SSH) est plus adaptée pour ce type de matériel et pour tous les projets susceptibles d'embarquer un Raspberry Pi (serveurs réseau, players multimédia, interfaçage et robotique, hacks avec les GPIO, etc.)

## 1 PI Cobbler & resistances

Lorsque vous raccordez le câble GPIO, assurez vous de **faire correspondre le fil rouge** avec la broche P1 du GPIO du Raspberry (du côté de la carte SD). **Le fil rouge, c'est la pin/broche #1 du câble**.
L'autre côté du câble ne peut s'insérer que d'une seule façon sur le Cobbler car il y a un détrompeur.

**a.** Voir TP_LED_raspberry_Pi

**b.** Raccorder une LED avec une **résistance limitant le courant** sur la broche #4 (**GPIO #4**) du Raspberry Pi.

**c.** Le script `raspi-pwm.py` fait varier par un signal PWM `(Puls With Modulation)` l'intensité de l'éclairement de la LED sur GPIO4.

## 2 web.py - Installation du serveur Web

Le Raspberry Pi est connecté en réseau par un connecteur `Ethernet` et/ou une liaison `WIFI`. Il peut devenir un serveur Web permettant le contrôle distant de processus ou la transmission à distance de données à travers Internet.
Il existe plusieurs bibliothèque de site Web embarqué pour Raspberry Pi, l'ordinateur a cependant des ressources limitées, il est préférable d'utiliser un serveur peut gourmand en ressources processeur et éviter les médias trop volumineux qui satureraient la mémoire flash. La bibliothèque retenue dans cet exemple est webpy .

**a.** Installation du serveur pour piloter les ports gpio à partir d'un navigateur internet à partir d'un Smartphone ou d'un iPad.

Déplacez-vous jusqu'au **répertoire qui contient vos scripts** avec la commande `cd /home/pi/Python`

- `sudo apt-get update`
- `sudo wget http://webpy.org/static/web.py-0.37.tar.gz`
- `sudo tar zxf web.py-0.37.tar.gz` *pour décompresser l'archive téléchargée* - Vous pouvez au passage la supprimer elle ne servira plus.
- `cd web.py-0.37`
- `sudo python setup.py install`

web.py *n'est pas encore disponible pour la version 3 de python.*

1. Rasp-Hack-LED « Traduction par MCHobby (www.MCHobby.be) - Vente de kit et composants »
2. La librairie RPi.GPIO est maintenant installée par défaut sur Raspbian. Pour s'assurer d'avoir la dernière version :

Déplacez-vous jusqu'au **répertoire qui contient vos scripts** avec la commande `cd /home/pi/Python` **puis saisir en console** les lignes suivantes :
- `sudo apt-get update`
- `sudo apt-get install python-rpi.gpio python3-rpi.gpio`

To install the latest development version from the project source code library : http://sourceforge.net/p/raspberry-gpio-python/wiki/install/

Pour vos futurs projets : `sudo apt-get install python-dev python3-dev`
`sudo apt-get install python-pip python3-pip`

**b.** Maintenant il va falloir créer une arborescence :

- `cd /` *retour à la racine*
- `sudo mkdir webpyserver` *le dossier du serveur*
- `cd /webpyserver`
- `sudo mkdir templates` *contiendra les pages html dont* `index.html`
- `sudo mkdir static` *feuille de style* CSS *éventuelle*

Le dossier `webpyserver` contiendra le fichier Python à exécuter ; le dossier `webpyserver/templates` contiendra une page HTML et le dossier `webpyserver/static` contiendra une éventuelle feuille de style au format CSS.
On va créer un programme nommé **gpio4.py dans le dossier** `webpyserver` qui permettra d'allumer ou d'éteindre une LED.
Le port du serveur Web sera 8080.

## 3 Le script `Python`

Ce programme récupère les données des GPIO et gère l'interface homme-machine de la page Web.

**a.** Déplacez-vous jusqu'au **répertoire** `/webpyserver` puis créez le fichier `gpio4.py`[3] ⇝ voir la documentation web.py

Listing 1 – gpio4.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import web
import RPi.GPIO as GPIO
from web import form
# definit GPIO4 en sortie
GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.OUT)
# definit la page de nom index pour le site web
urls = ('/', 'index')
dossier_web = web.template.render('templates/')
app = web.application(urls, globals())
# definit les boutons a afficher
ma_forme = form.Form(
form.Button("btn",id = "btnon",value = "on",html = "On",class_ = "bouton_on"),
form.Button("btn",id= "btnoff",value = "off",html = "Off",class_ = "bouton_off")
)
# definit l'action a effectuer quand la page index est appelee
class index:
    # quand la page est demandee
    def GET(self) :
        forme = ma_forme()
        return dossier_web.index(forme, "Raspberry Pi control GPIO4")
   # quand une forme web est soumise
    def POST(self):
        userdata = web.input()
        if userdata.btn == "on":
            GPIO.output(4,True)  # Allume la LED
        if userdata.btn == "off":
            GPIO.output(4,False) # Eteind la LED
    raise web.seeother('/')      # recharge la page web

# programme
if __name__ == '__main__':
    app.run()
```

**b.** **Éditer le fichier**.
Si vous aviez besoin de modifier `gpio4.py`, vous pouvez utiliser `nano` . . . rudimentaire mais efficace.

**c.** **Rendre le fichier exécutable**
Par défaut, les fichiers sont considérés comme des fichiers texte non exécutable . . . même s'ils contiennent des scripts.
Il faut donc indiquer au système d'exploitation qu'il peut autoriser l'exécution `gpio4.py`

```
chmod +x gpio4.py
```

3. Le guide ultime et définitif sur la programmation orientée objet en Python à l'usage des débutants qui sont rassurés par les textes détaillés qui prennent le temps de tout expliquer.

## 4 La page HTML

Dans le sous **dossier** `templates` créer le fichier `index.html` [4].
C'est la page d'accueil du site Web, elle est ici simplifiée au maximum. Il est possible de l'enrichir avec un éditeur HTML.

Listing 2 – index.html

```
$def with (form, title)
<!DOCTYPE html>
<html>
    <head>
        <title>$title</title>
    </head>

    <body>
        <br/>
        <form class="form" method="post">
        $:form.render()
        </form>
    </body>
</html>
```

## 5 Démarrer le serveur Web et accéder au serveur du Raspberry Pi depuis un ordinateur en réseau

**a.** Lancer le programme `gpio4.py` pour démarrer le serveur Web

**b.** À partir d'un navigateur de taper l'adresse IP du Raspberry en utilisant le port 8080 ;

XXX.XXX.XXX.XXX:8080
adresse IP du Rpi

Voilà le résultat à partir d'un iPad :


L'utilisation d'un module peut provoquer l'apparition d'un fichier d'extension `.pyc` . Il correspond à **la traduction du module en `bytecode` plus rapidement exploitable par l'interpréteur** Python. Ce fichier est généré à chaque modification du module. Lorsqu'un module est importé, Python vérifie la date des deux fichiers d'extension `.py` et `.pyc`. Si le premier est plus récent, le second est recréé. Cela permet au programme principal d'utiliser toujours la meilleure version du module.

## 6 Installation de WebIOPi une petite appli qui permet de contrôler les broches GPIO du RPi depuis une interface Web.

À partir du terminal, ou via votre client SSH lancer les commandes suivantes :

- wget http ://lycee.lagrave.free.fr/isn/raspberry/WebIOPi-0.7.0.tar.gz [5]
- tar xvzf WebIOPi-0.7.0.tar.gz
- cd WebIOPi-0.7.0
- sudo ./setup.sh

Vous êtes prêt à utiliser WebIOPi ! Mais le serveur et l'état GPIO sont perdus quand vous arrêter le script (`CTRL -C`) ou fermer le terminal.

**a.** Pour **démarrer** [6] **/ arrêter** le service de fond : `sudo /etc/init.d/webiopi start` **et** `sudo /etc/init.d/webiopi stop`

**b.** Ouvrir un navigateur et rentrée l'url : XXX.XXX.XXX.XXX:8000 sur n'importe quel appareil de votre réseau.
adresse IP du Rpi

**c.** Lors de la connexion les identifiants par défaut sont : • Utilisateur : « `webiopi` » • mot de passe : « `raspberry` »

**d.** **En choisissant le lien en-tête GPIO** sur la page principale, vous serez en mesure de contrôler GPIO avec une interface Web.
Cliquez sur une touche OUT/IN pour changer de direction GPIO.
Cliquez sur une broche pour changer l'état de sortie GPIO.

4. Toute page HTML peut être enrichie de formulaires interactifs, qui invitent vos visiteurs à renseigner des informations : saisir du texte, sélectionner des options, valider avec un bouton . . . tout est possible !
5. Looking for the latest version ?
6. Pour lancer WebIOPi au démarrage du système : `sudo update-rc.d webiopi defaults`

## 7 Début de projet … Interface Web simple et intuitive pour le Raspberry Pi

Ici on utilise un Raspberry Pi
mais pourquoi pas …
un projet sur Arduino ?

*Pour ce début de projet, il s'agit de contrôler des LED directement à partir du Raspberry Pi, puis à partir de n'importe quel appareil en réseau … votre Smartphone ou votre tablette.*

### a. Installation et utilisation de la bibliothèque WiringPi

Pour installer la librairie :

— `sudo apt-get install git-core`
— `git clone git://git.drogon.net/wiringPi`
— `cd wiringPi/`
— `git pull origin`
— `./build`

Vérifier l'installation avec la liste des pins/GPIO, leur état …

— `gpio readall`

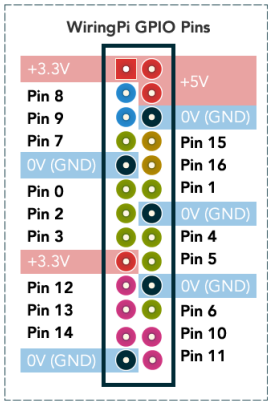
*Pour utiliser le numéro de broche réel (GPIO-4) à la place de celui de WiringPi (7 correspondant à GPIO-4), utiliser le drapeau `-g` dans votre commande.*

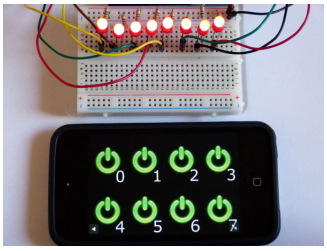
Selon l'explication donnée dans le manuel (`man gpio`), pour éteindre la LED reliée à la broche #4 (**GPIO #4**) du RPi puis la rallumer :

— `gpio -g mode 4 out` [8]
— `gpio -g write 4 1`
— `gpio -g write 4 0`

**Bien relier chaque LED à la bonne broche** [7] sur le GPIO du Raspberry Pi.

*La sortie 7 de la numérotation WiringPi correspond à GPIO 4 (le numéro de GPIO n'est pas sa position sur le connecteur, mais son numéro dans les registres de la puce ARM BCM2835), ou encore à la 8e patte en comptant de droite à gauche et de haut en bas depuis la SDcard.*



⇝ Pour contrôler votre RPi avec SSH. Vous pouvez utiliser Putty sous Windows ou ServerAuditor pour votre Smartphone.

### b. Installation d'un serveur web et mise en place du site web

Dans notre cas, nous n'avons pas besoin d'une base de données MySQL, juste d'un serveur HTTP et de son extension PHP.
Faire la mise à jour de votre RPi avec l'habituelle commande : `sudo apt-get update && sudo apt-get upgrade`
Installer le serveur HTTP Apache et l'extension PHP5 :

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```

Tester si votre serveur fonctionne en tapant l'adresse IP de votre RPi dans votre navigateur. Vous devriez maintenant voir une page indiquant « It works ! » ainsi que deux autres lignes. Si vous n'obtenez pas ça, alors vérifiez l'adresse IP de votre carte, essayez de réinstaller Apache ou redémarrer votre RPi. Cette page affiche que votre serveur Apache fonctionne correctement mais ne donne pas d'information sur son extension PHP. Pour vérifier cette dernière, accédez à votre répertoire `/var/www/` en utilisant la commande :

```
cd /var/www/
```

Avec la commande `ls`, vous devriez voir un seul fichier nommé `index.html` qui affiche « It works ! ». Vous pouvez maintenant le supprimer (`sudo rm index.html`) et en créer un autre appelé `index.php` (`sudo nano index.php`) avec le script suivant :

```
<?php
phpinfo();
?>
```

En actualisant votre navigateur, vous obtenez une page très longue, avec des informations sur votre serveur et sur PHP.
Si vous ne la voyez pas, vérifiez votre fichier `index.php`, essayez de réinstaller PHP ou essayez de comprendre l'erreur affichée à la place de la page attendue (effectuez une recherche sur Google si nécessaire).
Vous avez maintenant un serveur Apache/PHP entièrement fonctionnel.

7. En effet, pour WiringPi **les numéros de broches utilisés ne sont pas les mêmes** que ceux qui figurent sur le Raspberry Pi.
8. On peut utiliser de manière équivalente `gpio mode 7 out` pour une LED reliée à la broche #4 (**GPIO #4**).

**c. Transfert des fichiers de votre ordinateur à votre Raspberry Pi**

On peut installer un serveur FTP, mais ce n'est pas nécessaire, on peut transférer des fichiers en utilisant le protocole SFTP. Tout ce dont vous avez besoin est un client SFTP sur votre ordinateur [9].

Pour éviter des problèmes comme « *accès refusé* » ou « *écriture interdite* » car le propriétaire du répertoire `www` n'est pas **pi**, (en effet, si vous essayez la commande `ls -l /var/www`, vous verrez que seul `root` (le super-utilisateur) est propriétaire.) on modifie les droits :

```
sudo chown -R pi /var/www
```

Votre serveur est prêt à travailler et à recevoir des pages Web utilisant les langages HTML, CSS, JavaScript [10] et PHP.

**d. Contrôle de LED en PHP**


PHP signifie « PHP : Hypertext Preprocessor », c'est un langage de script côté serveur.
Cela signifie que le code PHP est exécuté par le serveur (à chaque fois que la page est demandée) et qu'il n'est pas visible par le client.

Éditez le fichier `index.php` [11] et insérez y le code suivant pour allumer la broche #4 (**GPIO #4**) :

```
<?php
system("gpio mode 7 out");
system("gpio write 7 1");
exec("gpio read 7", $status ;
print_r($status);
?>
```

**e. L'interface WEB :** Les ressources à télécharger


L'extension `.php` du fichier « `index.php` » permet au serveur de savoir qu'il y a du code PHP à exécuter avant d'envoyer la page générée contenant les 8 boutons. Ces boutons sont d'abord générés avec un `exec` dans une boucle. Ensuite, on détecte le moment où l'utilisateur clique sur un de ces boutons.

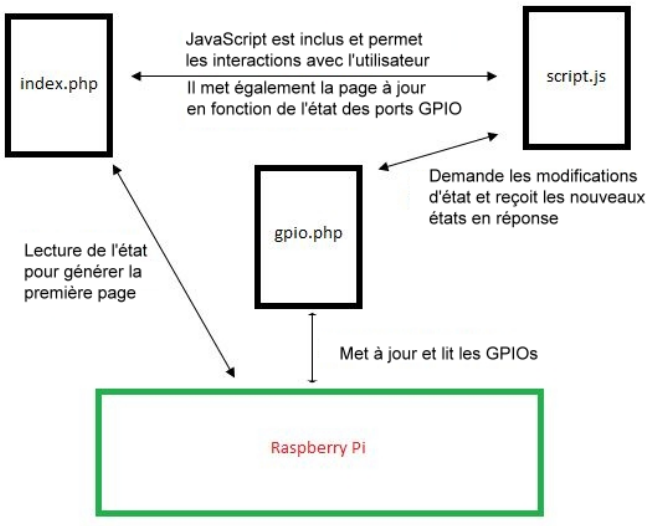
C'est là que le JavaScript est utile en ajoutant un « écouteur d'événement » à chacun des huit boutons et chaque fois que l'un d'eux est pressé, il utilise une fonction qui appelle `gpio.php`, reçoit sa réponse, puis la renvoie.

En fonction de cela, le JavaScript change le bouton rouge (pour OFF) en vert (pour ON). La page `gpio.php` contient le code PHP pour allumer/éteindre une LED en fonction de ce qu'envoie la fonction JavaScript.

L'utilisateur ne doit normalement pas appeler cette page directement, mais il y a une règle d'or lors de la création des sites Web et vous devez toujours vous rappeler ceci : « **Ne faites jamais confiance à l'utilisateur** ». En d'autres termes, ne pensez jamais que l'utilisateur va toujours faire ce que vous pensez qu'il va faire.

Il y a donc quelques sécurités au début du code PHP pour s'assurer que l'utilisateur passe une valeur correcte et pas une lettre par exemple.

Ci-contre un petit diagramme pour résumer tout ce texte.




9. WinSCP pour Windows ou Filezilla disponible à la fois pour Windows, pour MacOS et pour Linux.
10. Voir le document ISN_HTML_ex_Javascript_PHP
11. Voir en PHP : la fonction `system` et la fonction `exec`.