

Capture d'images et vidéo automatisée et Journalisation dans un serveur distant

Contenu

Objectifs de TP :	42
1. Utilisation d'une caméra avec la raspberry Pi :	42
1.1. Branchement et mise en œuvre :	42
1.2. Premier test du module picamera :	43
2. Service de capture d'images utilisant un archivage vers un serveur FTP:	48
2.1. Préparation d'un serveur FTP	48
2.2. Capture d'image et archivage sur un serveur FTP distant (FTP : File Tranfert Protocol).....	49
3. Service de commande supervision de GPIO et de caméra via vers un serveur web:	51
3.1. Préparation de la raspberry pi pour offrir un service web :	51
3.2. Création d'une interface web pour contrôler/superviser quelques GPIO et la picamera.....	52
4. Références:	57

Objectifs de TP :

- Capitaliser le savoir- faire acquis dans les TP1 et TP2 pour le contrôle des GPIO et le combiner avec l'utilisation de caméra pour automatiser des tâches de capture d'images et/ou vidéos.
- Exploiter l'intégration de la pile de protocoles TCP/IP dans linux embarquée pour permettre de journaliser les images capturées vers un serveur distant.
- Contrôler et superviser les entrées GPIO et caméra via une interface web

1. Utilisation d'une caméra avec la raspberry Pi :

1.1. Branchement et mise en œuvre :

La raspberry pi dispose d'un connecteur dédié à un module picamera spécifique pour la raspberry.

- Branchez le module picamera sur votre raspberry ; La photo de la figure 1, le branchement correct du module caméra sur la raspberry pi.

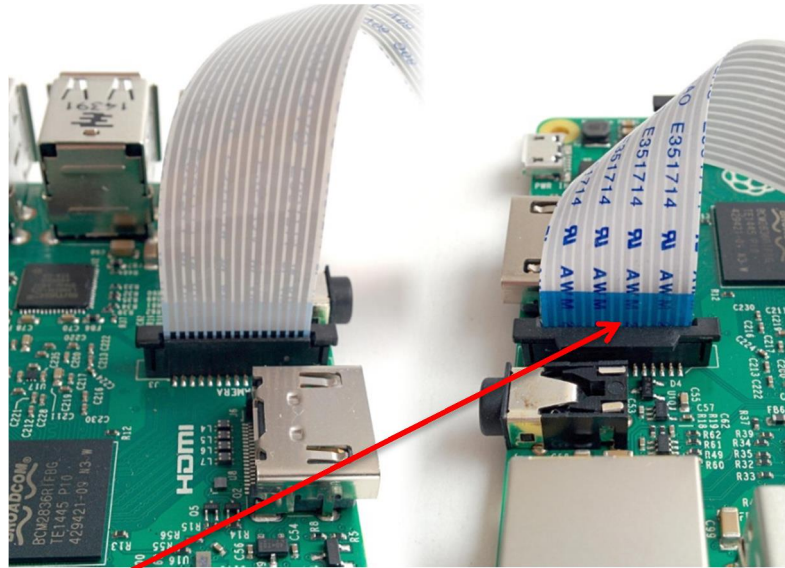


Figure 1. Branchement correct du module picamera sur la raspberry pi.

NB : le ruban bleu doit être du côté de connecteur Jack audio/video.

- Activez le module caméra à partir d'outil de configuration de la raspberry pi

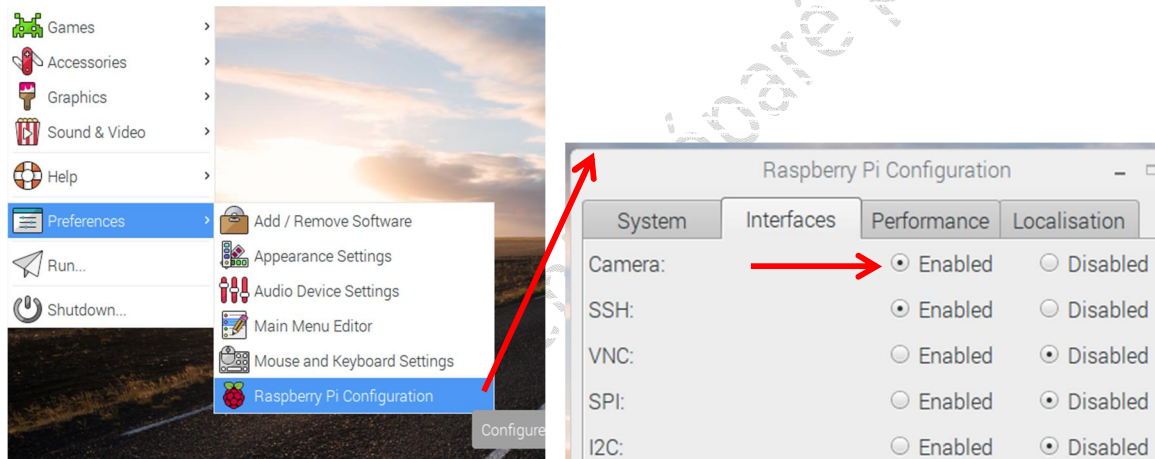


Figure 2. Activation du module picamera sur la raspberry pi

1.2. Premier test du module picamera :

Il existe une librairie écrite en python qui permet de commander le module picamera valable pour python 2.7 et 3.2 ; cette librairie s'appelle picamera :

référence : <https://picamera.readthedocs.io/en/release-1.13/>).

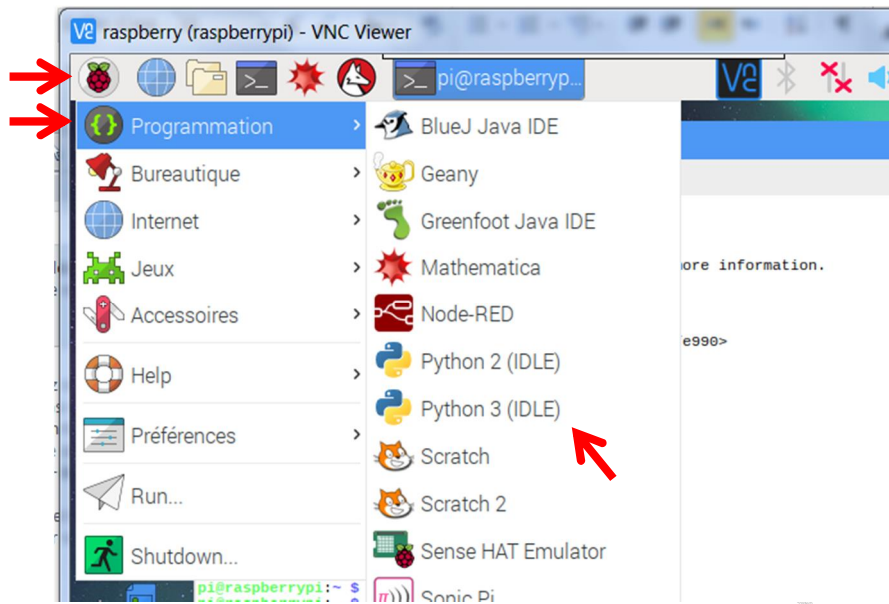
- Ouvrez un terminal et lancez l'interpréteur python ; importez le module picamera et essayez de prendre une photo le module picamera :

```
pi@raspberrypi:~ $ python
Python 2.7.13 (default, Nov 24 2017, 17:33:09)
[GCC 6.3.0 20170516] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from picamera import PiCamera
>>> from time import sleep
>>> Cam=PiCamera()
>>> Cam.start_preview()
<picamera.renderers.PiPreviewRenderer object at 0x769fe990>
>>> sleep(10)
>>> cam.capture('/home/pi/Desktop/im1.jpg')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'cam' is not defined
>>> Cam.capture('/home/pi/Desktop/im1.jpg')
>>> Cam.stop_preview()
>>> exit()
pi@raspberrypi:~ $ cat ./Desktop
cat: ./Desktop: est un dossier
pi@raspberrypi:~ $ ls ./Desktop
im1.jpg image1.jpg image2.jpg video.h264
pi@raspberrypi:~ $
```

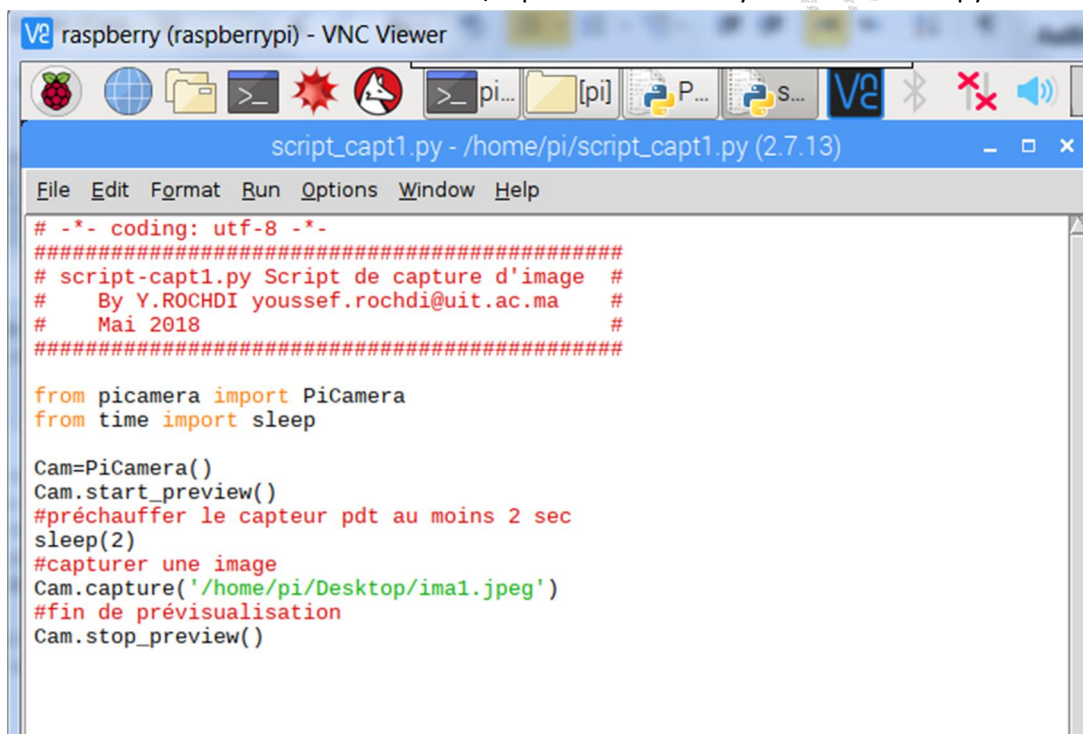
NB : Si vous êtes connectés à la raspberry via VNC Wiewer, vous ne pouvez pas voir la prévisualisation sur votre PC, il faut avoir un écran HDMI directement connecté sur la raspberry pi pour voir la prévisualisation.

La prévisualisation pendant une durée d'au moins 2 secondes est nécessaires pour chauffer le capteur de la caméra.

- Vérifiez la création de l'image sur le bureau, double-cliquez pour la visualiser.
- Lancez l'environnement de développement intégré IDLE pour Python :



- Créez un nouveau fichier : Menu File / Option New File et y saisissez le code python suivant :

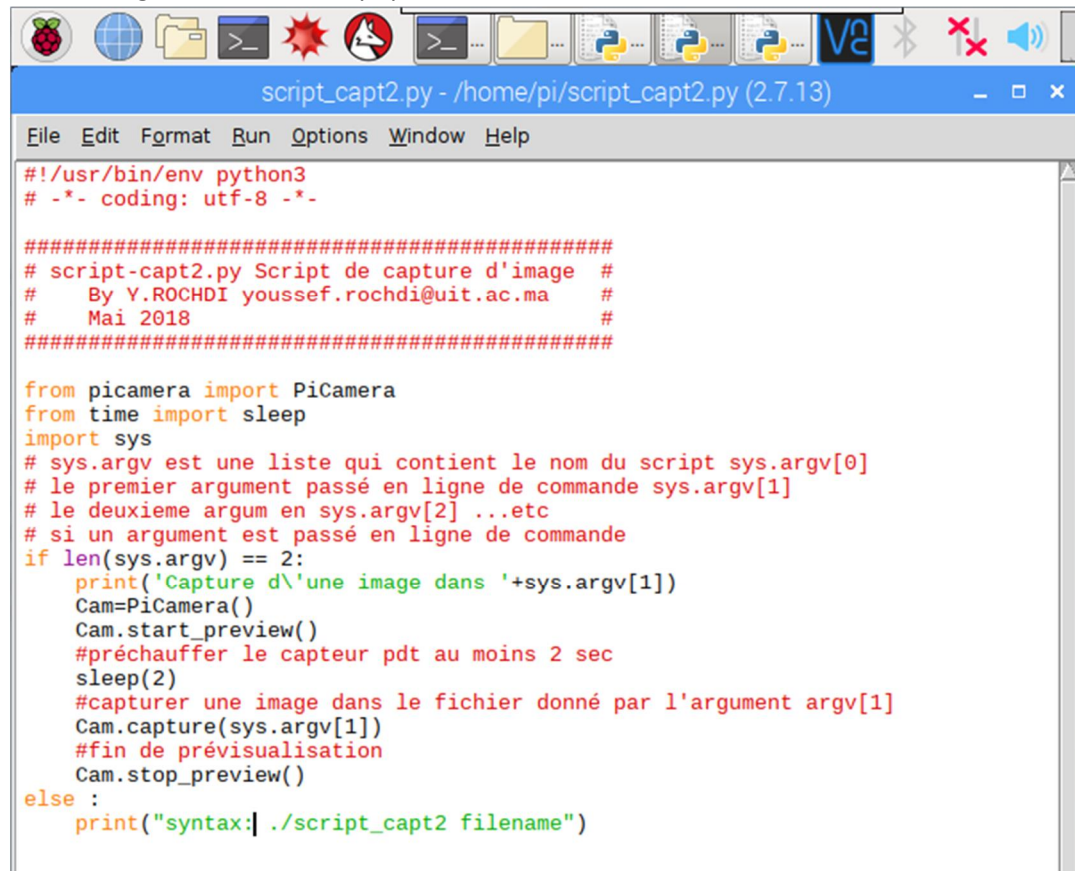


- Sauvegardez (File /Save As... ou Ctrl-s) sous le nom script_capt1.py et exécutez le (Run /Run Module ou F5) ; Vérifiez la création du fichier image ima1.jpeg sur le bureau et le visualisez par un double click.
- Le script précédent peut-être exécuté à partir d'un interpréteur de commande (Shell) ; **mais avant ça fermer IDLE et essayez la commande suivante :**

```
pi@raspberrypi:~ $ pyhton ./script_capt1.py
```

- Vérifiez qu'une nouvelle image a été capturée en visualisant le fichier ima.jpeg sur le bureau. Le problème avec cette commande est qu'à chaque fois qu'elle est lancée elle écrase l'ancienne image parce qu'elle écrit toujours dans le même fichier ima1.jpeg. On peut

dépasser ce problème en spécifiant un argument en ligne de commande indiquant le nom du fichier image. Modifiez le script précédent comme suit :



```

script_capt2.py - /home/pi/script_capt2.py (2.7.13)
File Edit Format Run Options Window Help

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

#####
# script-capt2.py Script de capture d'image #
# By V.ROCHDI youssef.rochdi@uit.ac.ma #
# Mai 2018 #
#####

from picamera import PiCamera
from time import sleep
import sys
# sys.argv est une liste qui contient le nom du script sys.argv[0]
# le premier argument passé en ligne de commande sys.argv[1]
# le deuxième argument en sys.argv[2] ...etc
# si un argument est passé en ligne de commande
if len(sys.argv) == 2:
    print('Capture d\'une image dans '+sys.argv[1])
    Cam=PiCamera()
    Cam.start_preview()
    #préchauffer le capteur pdt au moins 2 sec
    sleep(2)
    #capturer une image dans le fichier donné par l'argument argv[1]
    Cam.capture(sys.argv[1])
    #fin de prévisualisation
    Cam.stop_preview()
else :
    print("syntax:| ./script_capt2 filename")
  
```

- Rendez le script exécutable et le testez :

```

pi@raspberrypi:~$ chmod +x script_capt2
pi@raspberrypi:~$ ./script_capt2.py /home/pi/Desktop/ima3.jpeg
  
```

1.3. Capture d'image conditionnée par un événement externe ou chronométrique:

- Le script suivant permet de capturer une image chaque fois qu'une entrée GPIO passe à 0 ; Modifiez le script précédent comme suit et l'enregistrez sous le nom script_capt3.py :

```
script_capt3.py - /home/pi/script_capt3.py (2.7.13)
File Edit Format Run Options Window Help

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#####
# script-capt3.py Script de capture d'image #
# lorsqu'1 Gpio passe à 0 #
# By Y.ROCHDI youssef.rochdi@uit.ac.ma #
# Mai 2018 #
#####
from picamera import PiCamera
from time import sleep
import RPi.GPIO as GPIO
import sys
GPIO.setmode(GPIO.BCM)
GPIO.setup(0, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(26, GPIO.IN, pull_up_down=GPIO.PUD_UP)
compteur=1
Cam=PiCamera()
Cam.start_preview()
sleep(2)
while True:
    #si GPIO 0 passe à 0 capture de ima(compteur).jpeg
    if GPIO.input(0)==0:
        Cam.capture('./imag'+str(compteur)+'.jpeg')
        compteur+=1
        while GPIO.input(0)==0 : #tant que GPIO 0 est à 0
            if GPIO.input(26)==0: #si GPIO 26 passe à 0 fin du script
                Cam.stop_preview()
                sys.exit()
            else: #si GPIO 0 est à 1 tster si GPIO 26 passe à 0
                if GPIO.input(26)==0:
                    Cam.stop_preview()
                    sys.exit() #fin du script
```

- Lancez le script précédent, et dans un terminal provoquez un changement d'état de l'entrée GPIO.0 plusieurs fois en répétant la séquence de commande suivante :

```
pi@raspberrypi:~$ gpio -g mode 0 down; sleep 1 ; gpio -g mode 0 up
```

puis terminer le script en faisant passer GPIO.26 à 0 par:

```
pi@raspberrypi:~$ gpio -g mode 26 down
```

- Vérifiez que dans le répertoire où se trouve le script, il y a eu création d'autant de fichiers image (imag1.jpeg, imag2.jpeg, imag3.jpeg, ...) que de fois où GPIO.0 est passé à 0.
- Le problème avec le script précédent est que s'il est arrêté et relancer les premières images capturées seront écrasées ; puisque le compteur recommence toujours depuis 1. On peut résoudre ce problème en suffixant les noms de fichiers de capture par la date de capture, le script devient alors :

```

script_capt4.py - /home/pi/script_capt4.py (2.7.13)
File Edit Format Run Options Window Help

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#####
# script-capt4.py Script de capture d'image #
# lorsqu'un Gpio passe à 0, fichiers datés #
# By Y.ROCHDI youssef.rochdi@uit.ac.ma #
# Mai 2018 #
#####
from picamera import PiCamera
from time import sleep
import RPi.GPIO as GPIO
import sys
import datetime
def stamper():
    date=datetime.datetime.now()
    stamp=str(date.year)+'_'+str(date.month)+'_'+str(date.day)+'\
    '+'_'+str(date.hour)+'_'+str(date.minute)+'_'+str(date.second)
    return stamp
GPIO.setmode(GPIO.BCM)
GPIO.setup(0, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(26, GPIO.IN, pull_up_down=GPIO.PUD_UP)
compteur=1
Cam=PiCamera()
Cam.start_preview()
sleep(2)
while True:
    #si GPIO 0 passe à 0 capture de ima(compteur).jpeg
    if GPIO.input(0)==0:
        Cam.capture('./imag'+stamper()+'.jpeg')
        compteur+=1
        while GPIO.input(0)==0 : #tant que GPIO 0 est à 0
            if GPIO.input(26)==0: #si GPIO 26 passe à 0 fin du script
                Cam.stop_preview()
                sys.exit()
            else: #si GPIO 0 est à 1 tster si GPIO 26 passe à 0
                if GPIO.input(26)==0:
                    Cam.stop_preview()
                    sys.exit() #fin du script

```

- Vérifiez le bon fonctionnement du script, comme suit :

```

pi@raspberrypi: ~
Fichier Édition Onglets Aide

pi@raspberrypi:~$ ls
2018-05-05-195353_800x600_scrot.png  imag3.jpeg  Pictures  script_capt4.py
Desktop                               imag4.jpeg  Public    Templates
Documents                             imag5.jpeg  python_games  Videos
Downloads                             imag6.jpeg  script_capt1.py
imag1.jpeg                            Music       script_capt2.py
imag2.jpeg                            oldconffiles  script_capt3.py

pi@raspberrypi:~$ gpio -g mode 0 down ; sleep 1 ; gpio -g mode 0 up
pi@raspberrypi:~$ gpio -g mode 0 down ; sleep 1 ; gpio -g mode 0 up
pi@raspberrypi:~$ gpio -g mode 0 down ; sleep 1 ; gpio -g mode 0 up
pi@raspberrypi:~$ gpio -g mode 26 down
pi@raspberrypi:~$ ls
2018-05-05-195353_800x600_scrot.png  imag2.jpeg  Public
Desktop                               imag3.jpeg  python_games
Documents                             imag4.jpeg  script_capt1.py
Downloads                             imag5.jpeg  script_capt2.py
imag1.jpeg                            imag6.jpeg  script_capt3.py
imag2018_5_5_20_40_35.jpeg            Music       script_capt4.py
imag2018_5_5_20_40_39.jpeg            oldconffiles  Templates
imag2018_5_5_20_40_41.jpeg            Pictures     Videos
pi@raspberrypi:~$

```

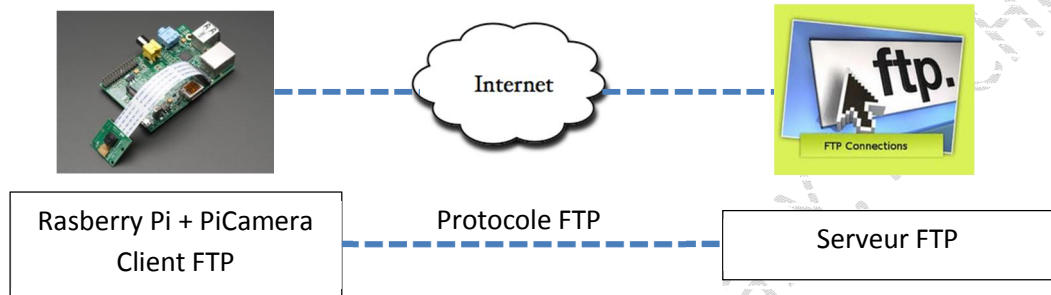
2. Service de capture d'images utilisant un archivage vers un serveur FTP:

2.1. Préparation d'un serveur FTP

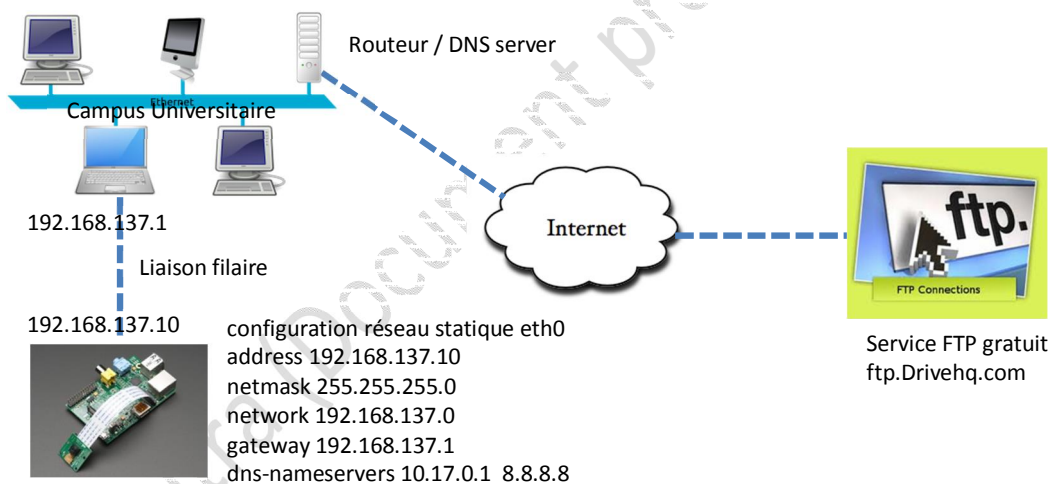
On se propose de modifier écrire un script capable de prendre une image chaque 5 min, chaque jour (à part les week end) entre 6h et 18h et de l'envoyer vers un serveur FTP pour

archivage. Ce genre de script peut être utile pour suivre par exemple le déroulement des travaux sur un chantier à distance, ou enregistrer l'évolution d'un phénomène dynamique comme la croissance d'une plante, ou observer le comportement d'un animal sans le déranger...etc

La topologie logique qu'on utilisera sera la suivante :



La topologie physique qu'on utilisera dans cette partie sera la suivante :



- Tout d'abord il faut disposer d'un serveur FTP ; utilisez le service FTP gratuit sur www.Drivehq.com : Il faut d'abord s'inscrire pour obtenir un compte (login et password).

2.2. Capture d'image et archivage sur un serveur FTP distant (FTP : File Tranfert Protocol)

- Python dispose déjà d'une librairie FTP client (pour plus de détails voir Références) ; Modifiez le script_capt4.jpeg en script_capt5.jpeg comme suit :


```

script_capt5.py - /home/pi/script_capt5.py (3.5.3)
File Edit Format Run Options Window Help

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#####
# script-capt5.py Script de capture d' une image #
# datée et envoyée vers un serveur FTP #
# By Y.ROCHDI youssef.rochdi@uit.ac.ma #
# Mai 2018 #
#####
from picamera import PiCamera
from time import sleep
import sys
import datetime
import ftplib
import os

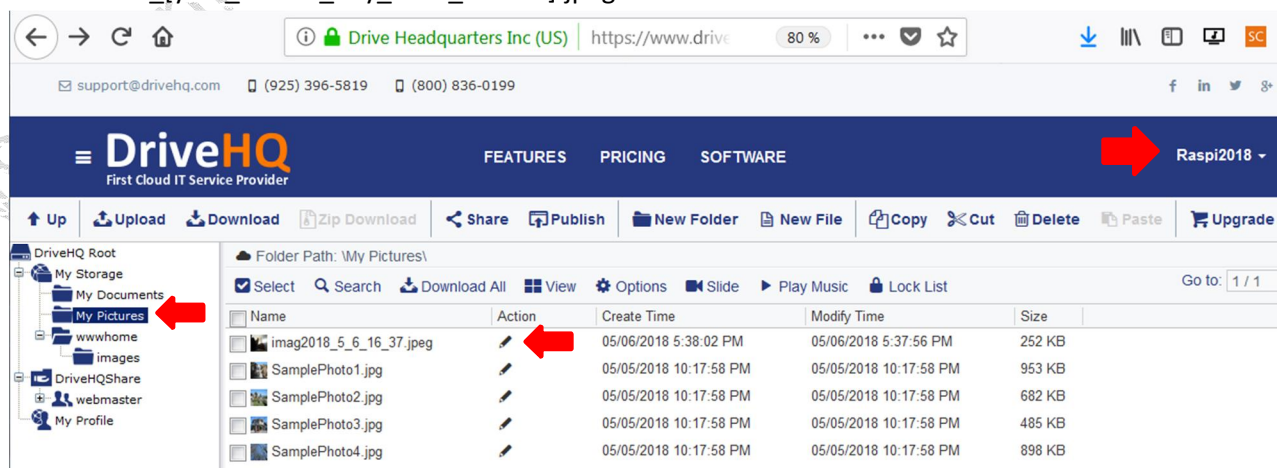
def stamper():
    date=datetime.datetime.now()
    stamp=chr(date.year)+'_'+chr(date.month)+'_'+chr(date.day)+'\
        '+chr(date.hour)+'_'+chr(date.minute)
    return stamp
#fonction de chargement (upload) de file vers serveur ftp
# option ==0 fichier texte copie linne by line
# autre fichier binaire copie de block de données binaires
def upload(ftp, file, option):
    if option ==0:
        ftp.storlines("STOR " + file, open(file))
    else:
        ftp.storbinary("STOR " + file, open(file, "rb"), 1024) #block de 1024
# connexion au serveur ftp en utilisant login et password
ftp = ftplib.FTP("ftp.Drivehq.com")
ftp.login("Raspi2018", " ")

#changer de répertoire dans le serveur pour enregistrer ds /My Pictures
ftp.cwd("/My pictures")
#Capture d'une image
Cam=PiCamera()
Cam.start_preview()
sleep(2)
file_name='./imag'+stamper()+'.jpeg'
Cam.capture(file_name)
Cam.stop_preview()
#transfert du fichier
upload(ftp, file_name, 1)
#suppression de la copie local (optionnel)
os.system("rm "+file_name)

```

A personnaliser selon votre compte chez Drivehq.com

- Exécutez le script (F5) ; attendez un peu et consultez le dossier My Pictures sur le serveur FTP (Il sera peut être nécessaire de changer de répertoire pour rafraîchir l’affichage) ; vous devez trouver un nouveau fichier ajouté portant le nom ima_[year_month_day_hour_minute].jpeg



- Rendez le script script_capt5.py exécutable et utilisez le service cron pour permettre de prendre une image chaque 2 minutes, chaque jour entre 6h et 18h sauf les week-ends.
- Testez d'abord votre script localement en mettant le transfert vers le serveur FTP (upload (...)) et aussi l'instruction de suppression à la fin en commentaires. Puis testez le avec le transfert vers le serveur FTP : au bout de 10 min le script aura capturé et transférer 5 images ; vérifiez le contenu de My Pictures sur le serveur FTP.

3. Service de commande supervision de GPIO et de caméra via vers un serveur web:

3.1. Préparation de la raspberry pi pour offrir un service web :

- Commencez par installer le serveur web Apache2 et le moteur php :

```
pi@raspberrypi:~$ sudo apt-get install apache2 php7.0
```

```
pi@raspberrypi:~$ sudo reboot
```

- A partir de votre PC connecté à la raspberry Pi via Ethernet sur le réseau 192.168.137.0, ouvrez un navigateur et tapez l'adresse 192.168.137.10 (Adresse de votre raspberry Pi). Vous devez avoir une page web indiquant que apache2 fonctionne (It's work...).
- Test du fonctionnement du moteur PHP : suppression du fichier index.html et création d'un fichier index.php dans /var/www/html, avec le contenu :

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

```
pi@raspberrypi:~$ sudo rm /var/www/html/index.html
```

```
pi@raspberrypi:~$ sudo nano /var/www/html/index.php
```

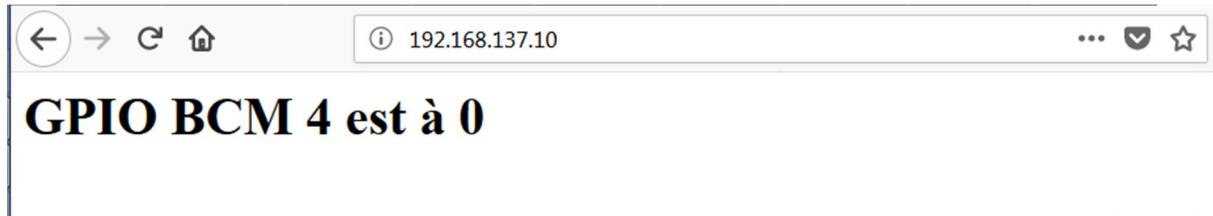
- Actualisez la page web ouverte sur le navigateur de votre PC, cette fois-ci vous devez avoir des informations (rôle de la fonction phpinfo()) sur le moteur php installé sur votre raspberry pi.
- On peut exécuter des commandes shell dans un script php à l'aide des fonctions system ou exec ; modifiez le fichier /var/www/html/index.php comme suit :

```
<h1>
<?php
system ("gpio -g 4 write 0");
exec ("gpio -g read 4", $status4);
echo ("GPIO BCM 4 est à " . $status4[0]);
?>
</h1>
```

NB : lorsque votre navigateur demande la page index.php au serveur web/php, ce dernier exécute le script php. Ce dernier, à travers les commandes shell met à 0 la GPIO4 (BCM) (GPIO4 doit être déjà configurée en sortie manuellement), puis lit l'état de cette GPIO, la sortie de la commande exec est stocké dans le tableau status4 les lignes de sortie de la commande, et finalement le code php génère

un texte par concaténation entre GPIO BCM 4 est à et le contenu de la variable \$status4[0] la première ligne affichée par la commande gpio read

- Actualisez à nouveau la page web ouverte dans votre navigateur du PC pour voir l'effet.



3.2. Création d'une interface web pour contrôler/superviser quelques GPIO et la picamera

- En utilisant un éditeur de texte graphique comme TextEditor (disponibles dans le menu des applications /rubriques Accessoires), créez sur le bureau, un nouveau fichier que vous allez index.php et y copiez le code HTML/PHP suivant :

```
<?php
if(!empty($_POST['GPIO4On'])) // si ON pour GPIO4 appuyé.
{
    system ( "gpio -g write 4 1" );
}
if(!empty($_POST['GPIO4Off'])) // si OFF pour GPIO4 appuyé.
{
    system ( "gpio -g write 4 0" );
}
if(!empty($_POST['Capturef'])) // Si faible resol appuyé.
{
    system ( "./capturef.py" );
}
if(!empty($_POST['Captureh'])) // Si haute resol appuyé.
{
    system ( "./captureh.py" );
}
?>

<h1 style="text-align: center;">Interfacage Web d'une Raspberry Pi</h1>
<h1 style="text-align: center;">GPIOs et camera</h1>
<p style="text-align: center;">Commander les sorties suivantes:</p>
<table
style=" background-color: #00ff00; margin-left: auto; margin-right: auto;">
<tbody>
<tr>
<td style="text-align: center; width: 124px;">GPIO Num (BCM)</td>
<td style="text-align: center; width: 50px;">4</td>
<td style="text-align: center; width: 50px;">5</td>
<td style="text-align: center; width: 50px;">6</td>
</tr>
<tr>
<td style="text-align: center; width: 124px;">Etats</td>
<td style="text-align: center; width: 50px;">
<?php
    exec ( "gpio -g read 4", $status4 );
    echo ( $status4[0] );
?>
</td>
<td style="text-align: center; width: 50px;"> ? </td>
<td style="text-align: center; width: 50px;"> ? </td>
</tr>
<tr>
<td style="text-align: center; width: 124px;">Mise à jour 1/0</td>
<td style="text-align: center; width: 50px;">
<form method="post" action="">
```

```

<input type="submit" name="GPIO4On" value="ON" />
<input type="submit" name="GPIO4Off" value="OFF" />
</form>
</td>
<td style="text-align: center; width: 50px;">
</td>
<td style="text-align: center; width: 50px;">
</td>
</tr>
</tbody>
</table>
<p style="text-align: center;">=====</p>
<p style="text-align: center;">Observer les &eacute;tat des entr&eacute;es
suivantes</p>
<table style="background-color: #00ff00; margin-left: auto; margin-right:
auto;">
<tbody>
<tr>
<td style="text-align: center; width: 124px;">GPIO Num (BCM)</td>
<td style="text-align: center; width: 50px;">13</td>
<td style="text-align: center; width: 50px;">19</td>
<td style="text-align: center; width: 50px;">26</td>
</tr>
<tr>
<td style="text-align: center; width: 124px;">Etat</td>
<td style="text-align: center; width: 50px;">
<?php
    exec ( "gpio -g read 13", $status );
    echo ( $status[0] );
?></td>
<td style="text-align: center; width: 50px;"> ?</td>
<td style="text-align: center; width: 50px;"> ?</td>
</tr>
</tbody>
</table>

<form method="post" action="index.php">
<p style="text-align: center;">
<input type="submit" name="Rafraichir" value="Rafraichir" />
</p>
</form>

<p style="text-align: center;">=====</p>
<p style="text-align: center;">Capture Camera &agrave; la demande</p>
<p style="text-align: center;">

</p>
<p style="text-align: center;">À modifier "./image.jpeg?time=<?php echo time(); ?>"
<form method="post" action="">
<p style="text-align: center;">
<input type="submit" name="Capturef" value="faible resol" />
<input type="submit" name="Captureh" value="haute resol" />
</form>
</p>
<p style="text-align: center;">
By Y.ROCHDI youssef.rochdi@uit.ac.ma
</p>
</p>

```

- Le fichier index.php créé combine du code HTML, PHP mais aussi fait appel à des scripts python : capturef.py et captutreh.py qui permettent de capturer une image respectivement en faible résolution ou en forte résolution, et de l'enregistrer sous le nom image.jpeg dans le même répertoire que les scripts. Dans l'éditeur IDLE de python 3 créez sur le bureau ces deux scripts avec les contenus suivants :


```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#####
# capturef.py Script de capture d'image #
# faible résolution
# By Y.ROCHDI youssef.rochdi@uit.ac.ma #
# Mai 2018 #
#####

from picamera import PiCamera
from time import sleep
import os
Cam=PiCamera()
Cam.resolution = (300,200)
Cam.start_preview()
#préchauffer le capteur pdt au moins 2 sec
sleep(2)
#capturer une image
Cam.capture('./image.jpeg')
#fin de prévisualisation
Cam.stop_preview()

```

NB : la seule différence entre les deux scripts est juste la résolution de prise d'image par la caméra.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
#####
# captureh.py Script de capture d'image #
# haute résolution
# By Y.ROCHDI youssef.rochdi@uit.ac.ma #
# Mai 2018 #
#####

from picamera import PiCamera
from time import sleep
import os
Cam=PiCamera()
Cam.resolution = (1024,768)
Cam.start_preview()
#préchauffer le capteur pdt au moins 2 sec
sleep(2)
#capturer une image
Cam.capture('./image.jpeg')
#fin de prévisualisation
Cam.stop_preview()

```

- Copiez les fichiers : index.php , capturef.py, captureh.py vers le répertoire de publication des pages web du serveur web apache /var/www/html/, attribuez les permissions correctes pour l'utilisateur www-data et le groupe www-data (utilisateur et groupe d'utilisateur ayant invoquer le serveur web apache) et rendre les scripts python exécutables :

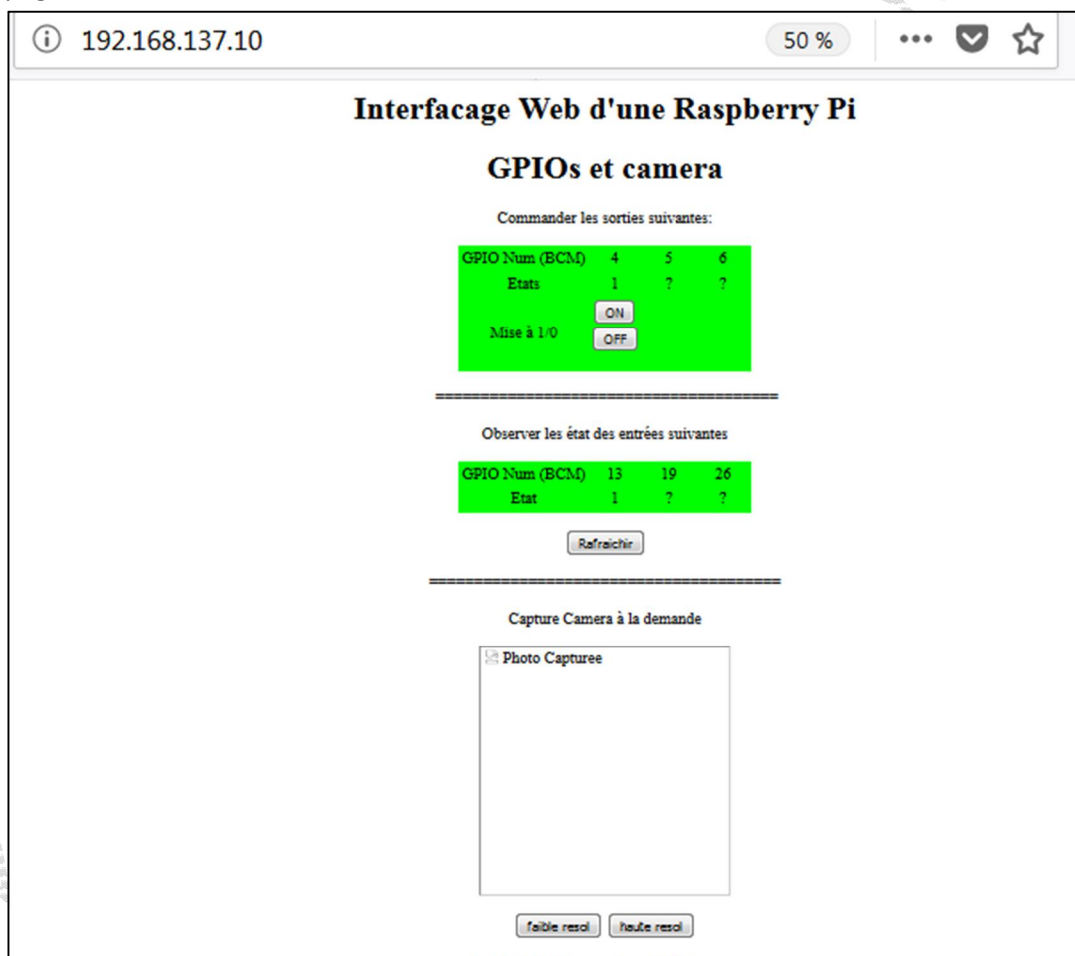
```
pi@raspberrypi:~ $ sudo cp ./Desktop/index.php ./Desktop/capturef.py ./Desktop/captureh.py \
```

```
> /var/www/html
pi@raspberrypi:~$ sudo chown -R www-data:www-data /var/www/html
pi@raspberrypi:~$ sudo chmod 775 /var/www/html/index.php /var/www/html/capturef.py \
> /var/www/html/captureh.py
pi@raspberrypi:~$ sudo reboot
```

- Préparez les sorties GPIO (4,5 et 6) et les entrées (13,19 et 26) en notation BCM :

```
pi@raspberrypi:~$ gpio -g mode 4 out
pi@raspberrypi:~$ gpio -g mode 13 in
pi@raspberrypi:~$ gpio -g mode 13 up
```

- Actualisez la page web ouverte dans le navigateur de votre PC, vous devez obtenir à une page web comme celle donnée ci-dessous :



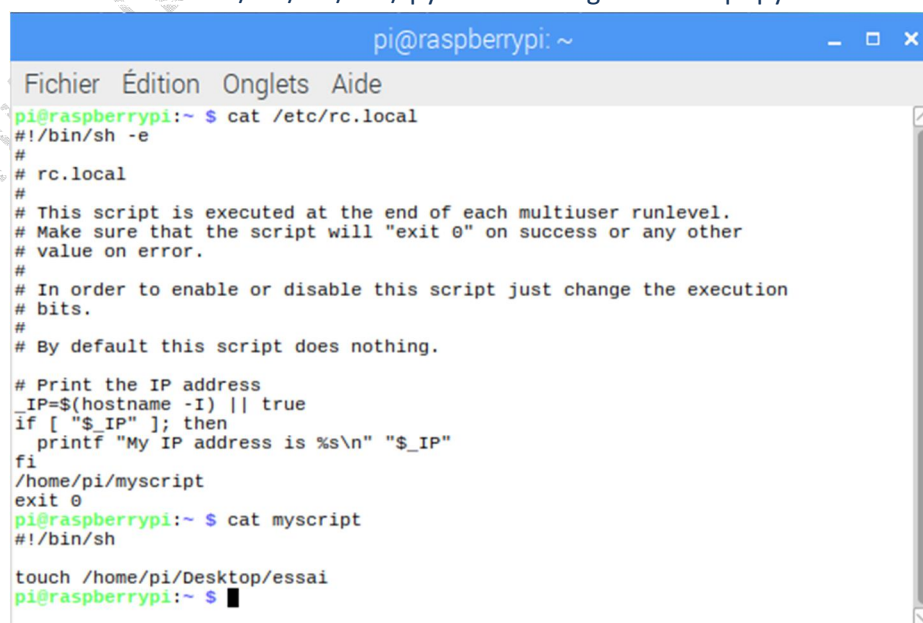
- Essayez de modifier l'état de la sortie GPIO4 à l'aide des boutons On et Off ; puis modifiez l'état de l'entrée 13 à l'aide de la commande `gpio -g mode 13 down` et cliquez sur le bouton rafraichir pour voir le changement rapporté sur la page web. Mais pour le moment aucun image n'est encore capturée, son cadre reste alors vide avec juste le texte alternatif Photo Capturée.
- Pour que l'utilisateur connecté via le serveur web (soit www-data) puisse accéder à la caméra il faut qu'il soit membre du groupe video ; pour cela ajoutez l'utilisateur www-data dans le groupe video :

```
pi@raspberrypi:~ $ sudo usermod -a -G video www-data
pi@raspberrypi:~ $ sudo reboot
```

- N'oubliez pas de reconfigurer les sorties et les entrées commandées/supervisées par la page web ; car à chaque redémarrage les états des GPIO sont réinitialisés.
- Rafraichir la page web de commande/supervision sur votre navigateur. Cliquez sur le bouton "faible resol", la page est actualisée avec une image capturée de faible résolution. Essayez l'autre bouton haute résolution.

Exercices :

1. Complétez le code de la page index.php pour permettre la modification des sorties 5 et 6, et l'observation de l'état des entrées 19 et 26.
2. Compléter la page web avec un bouton « Archiver » à côté des boutons haute resol et faible resol qui permet d'archiver la dernière image capturée vers le serveur FTP, le nom du fichier image doit être de la forme webima_année_mois_jour_heure_min_sec.jpeg .
3. En consultant la librairie picamera (voir documentation donnée en références), modifiez la page index.php pour ajouter un bouton de capture d'une vidéo en format mpeg, d'une durée de 10s puis l'envoyer vers le serveur FTP.
4. Lorsque la raspberry redémarre par panne de courant par exemple, les GPIO sont reconfigurées en INPUT ; donc le contrôle à partir de la page web des sorties 4,5 et 6 n'est plus possible (redémarrer la Rspi pour vérifier ce problème). Pour éviter, ce problème on peut créer un script qui configure les GPIO 4, 5 et 6 en sorties à chaque démarrage. Pour qu'un script « myscript » soit exécuté à chaque démarrage on doit suivre la procédure suivante :
 - créer le script (en langage shell ou python), puis le rendre exécutable.
 - modifier le fichier /etc/rc.local par nano (par exemple pour y ajouter avant exit 0 , le nom du script soit /chemin_d_access/myscript
 - N'oubliez pas d'ajouter au début du script #!/bin/sh s'il s'agit d'un script shell ou #!/usr/bin/env/ python3 s'il s'agit d'un script python.



```
pi@raspberrypi: ~
Fichier Édition Onglets Aide
pi@raspberrypi:~ $ cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi
/home/pi/myscript
exit 0
pi@raspberrypi:~ $ cat myscript
#!/bin/sh
touch /home/pi/Desktop/essai
pi@raspberrypi:~ $
```

L'exemple précédent, permet à chaque démarrage d'exécuter le script qui met à jour la date d'un fichier essai qui se trouve sur le bureau de l'utilisateur pi. La date de ce fichier nous renseigne alors sur la date de démarrage de la Rspi.

Exploitez cette technique pour résoudre le problème de configuration des GPIOs 4,5 et 6 en sorties à chaque démarrage, avec un script python puis par un script shell.

4. Références:

- Utilitaire gpio de la librairie wiringPi :
<http://wiringpi.com/the-gpio-utility/>
- Documentation de librairie FTP client en python 3 :
<https://docs.python.org/3/library/ftplib.html>
- Documentation de librairies picamera :
<https://picamera.readthedocs.io/en/release-1.13/index.html>
- Editeur de code HTML en ligne :
<https://html-online.com/editor/>
- Documentation de la librairie python pour l'utilisation des GPIO RPi.GPIO :
<https://sourceforge.net/p/raspberry-gpio-python/wiki/BasicUsage/>