

Mise en place d'un système de vidéo-Surveillance personnel

Table des matières

1 Présentation générale.....	3
1.1 Pré-requis.....	3
1.1.1 Matériels et Internet.....	3
1.1.2 Connaissances :.....	4
1.2 Les photos des composants :.....	5
2 Le plan de cablage des composants.....	13
2.1 Pour la partie raspberry : vidéo et éclairage pièce principale.....	13
2.2 Partie distante esp-8266-01	15
3 Mise en place du serveur Web sur le Raspberry 3.....	18
4 Mise en place des scripts Raspberry.....	19
4.1 Contrôle de la connexion wifi.....	21
4.2 Relance de la camera au reboot.....	21
4.3 Transfert automatique des photos et vidéos sur Dropbox.....	22
4.4 Mise en place d'un crontab user.....	23
5 Compte Dropbox.....	24
6 Gestion camera et nettoyage vidéo.....	26
7 Flashage des dispositifs ESP-8266-01.....	31

1 Présentation générale

Le but est de mettre en place un système de vidéo-surveillance accessible à distance depuis un navigateur web, donc à partir de :

- un micro-ordinateur
- une tablette numérique
- un téléphone mobile

Le dispositif détecte tout mouvement (package motion raspbian) et il déclenche la prise d'une photo et le démarrage d'un enregistrement vidéo de 30s. Les photos et les vidéos de plus de 1Mo sont envoyées toutes les 10s dans un compte [Dropbox](#).

Je m'appuierai sur ce qui existe déjà sur internet rn donnat les liens pour aider à cette mise en place.

Pour ma part j'ai réalisé ce montage avec une Livebox Play de chez Orange. A adapter avec les autres opérateurs.

Cout total autour de 90 à 100 €

1.1 Pré-requis

1.1.1 Matériels et Internet

- Un Raspberry pi 3 (qui a le wifi intégré) , son boitier d'alimentation et une micro-SD 16 ou 32 Go, un ensemble de cables
- Disposer d'un accès Internet à partir d'une Box.
- La Box devra disposer d'un accès depuis internet avec le mécanisme de DynDns.
 - Par exemple , vous trouverez [ici](#) comment obtenir un domaine internet gratuit chez [no-ip](#) (<https://www.noip.com/>) . Il existe d'autres fournisseurs ...
- La Box devra permettre le routage local à travers le mécanisme de [NAT/PAT](#) (<https://raspbian-france.fr/mettre-en-ligne-serveur-web-raspbian-dydns-port-forwarding/>) Sur Livebox Play pas de loopback (en local il faudra mettre l'URL locale ou son alias DNS fourni par la box)
- une [adresse fixe locale](#) (<http://pixelboys.fr/definir-une-ip-statique-pour-le-rpi-chez-orange/>) pour le Raspberry pi 3, bien noter l'alias DNS utilisé.
- Pour maintenance à distance, et une utilisation plus avancée, la Box devrait aussi permettre l'accès SSH pour accéder à toutes les fonctionnalités du Raspberry pi., (Installer [JuiceSSH](#) (<https://juicessh.com/>) sur votre tablette ou mobile et paramétrier

le NAT/PAT de votre Box pour le protocole SSH afin d'atteindre le raspberry). Voir le [lien](#) donné plus haut

- une camera module V2 pour Raspberry 3
- un moteur à pas pour permettre la rotation de la camera
- 2 relais 5V / 220V AC pour permettre l'allumage d'une lampe pour améliorer l'image caméra et/ ou simuler une présence et pour déclencher un arrosage.
- Pour un système complémentaire de déclenchement d'arrosage (ou autre) , un module ESP8266-01 et son interface de programmation USB (voir photos ci-dessous)

1.1.2 Connaissances :

J'ai réalisé ce montage à l'aide d'un Raspberry 3 dont le système d'exploitation est Raspbian (stretch Linux) , pour réaliser l'installation il faut avoir quelques connaissances linux :

- savoir se connecter/déconnecter sous son compte utilisateur **pi**
- connaître les principales commandes d'administration unix : **ls, cd, chmod, chown, cp, rm, mv, vi, nano, cat, sudo, su ...**
- savoir lancer des commandes **root** en utilisant **sudo** (installation de package, modification de certains fichiers de configuration apache, motion ...)
- savoir utiliser un éditeur de texte de type **geany** ou mieux **vi**.
- Pour avoir un accès plus convivial au raspberry :
 - Installer Win32DiskManager sur votre PC Windows
 - Installer le package **RealVNCClient** sur votre PC

Pour récupérer une image Raspbian et l'installer sur la carte micro-sd suivre ce tutorial :

Pour Windows : <https://raspbian-france.fr/creez-carte-sd-raspbian-raspberry-pi-windows/>

Pour Linux (sur votre PC, je suppose que vous avez les connaissances unix) réaliser avec la commande **dd** :

<https://raspbian-france.fr/creation-carte-sd-raspberry-raspbian-sous-gnulinux/>

Premier démarrage de Raspbian et configuration :

Pour la première configuration, si vous voulez y accéder depuis votre PC Windows à l'aide d'un cable RJ45 commencer par :

<https://easydomoticz.com/connecter-raspberry-directement-a-un-pc-windows/>

puis continuer la configuration par :

<https://raspbian-france.fr/installer-raspbian-premier-demarrage-configuration/>

Lors de la configuration :

Activer les connexions SSH :

<http://www.soft-alternative.com/raspberry-pi-activer-desactiver-serveur-ssh-raspbian.php>

Activer le serveur VNC :

<https://raspbian-france.fr/vnc-raspberry-pi/>

Activer le Wifi pour votre BOX (pré-requis connaître le réseau : Nom de la Box et son mot de passe SSID) :

<http://mars-hack-lab.fr/index.php/2018/02/09/configuration-de-la-connexion-wifi-raspbian/>

Gestion des packages ajout et mise à jour :

<https://angristan.fr/mettre-a-jour-raspberry-pi-sous-raspbian/>

<http://debian-facile.org/doc:système:apt:apt-get>

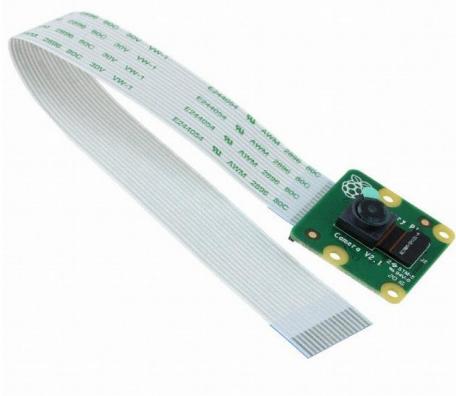
1.2 Les photos des composants :



Raspberry pi 3



Le boîtier, alimentation, refroidissement (ici boîtier Smraza)



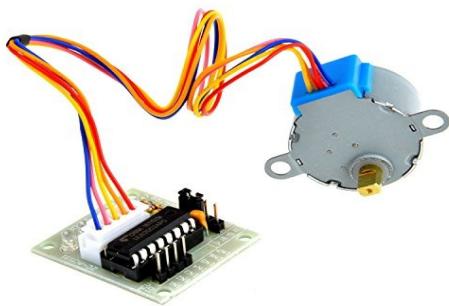
Le module camera HD



La Micro-SD 16GB font l'affaire



Ensemble de câbles



Le moteur à pas et carte de commande 5V 28BYJ-48



2 Relais 2 voies pour activer lumière ou autre



ESP-8266-01, module wifi (en prévoir 2 ou 3 pour quelques

euros)



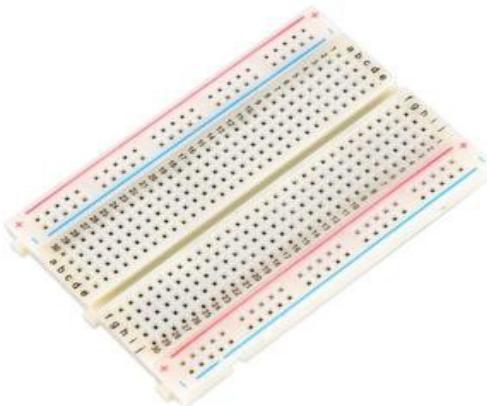
Pinzhi ESP8266 ESP-01 Flasher mod.prog, USB Programmeur

Uploader pour Arduino IDE, IoT



Alimentation 5V 3,3V Yurobot (pour dispositif éloigné :

déclenchement arrosage dans mon cas)



Carte Breadboard pour dispositif éloigné.



Un ensemble Prise Malle/femelle et prise coupure

Male avec terre

Femelle avec terre

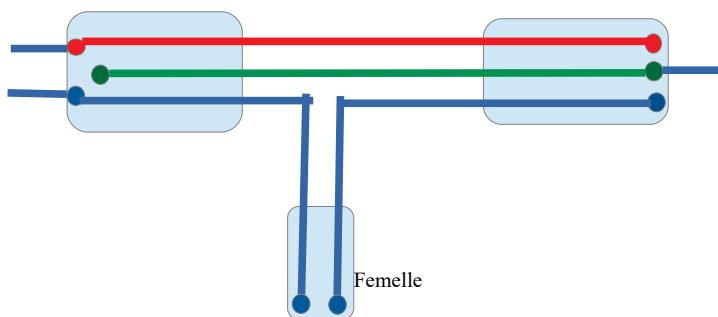


Schéma de câblage du Té

Puis pour assembler trouver une boite plastique pour fixer les composants. La photo ci-dessous montre à quoi ressemble ce dispositif bricolé pour la partie vidéo et allumage d'une lampe :



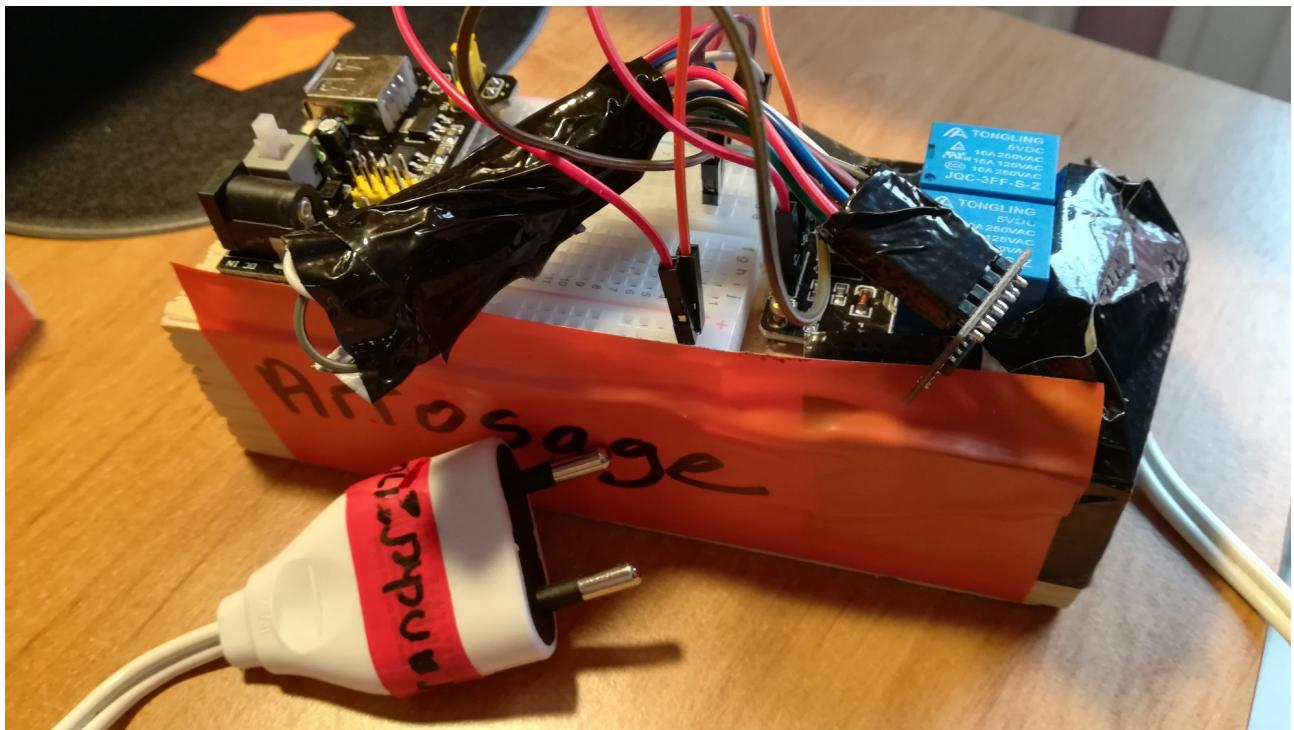
Si on désolidarise la lampe du raspberry (option 2) du relais de l'allumage lampe, on a unmontage rasberry/camera avec moteur à pas pour pivoter qui ressemble à l'image ci-dessous.



Pour la partie allumage :

- de l'arrosage par shunt d'un interrupteur (option 1) par dispositif distant
- et pour la partie allumage de lampe dans l'option 2 par dispositif distant

on a un montage identique , seul le flash de l'ESP8266 est différent dans l'URL testée :
présence ou pas du fichier **arrosage.cam** (respectivement **lumiere.cam** ; voir plus loin dans
le document) :



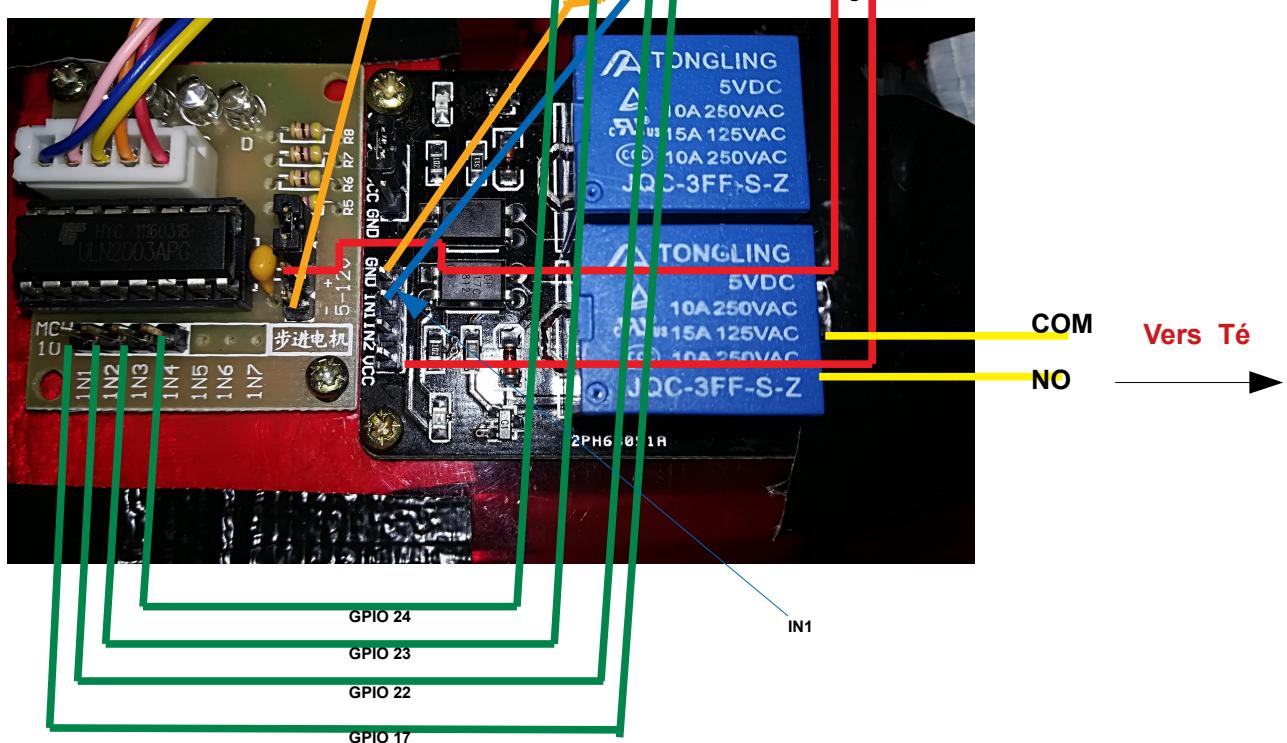
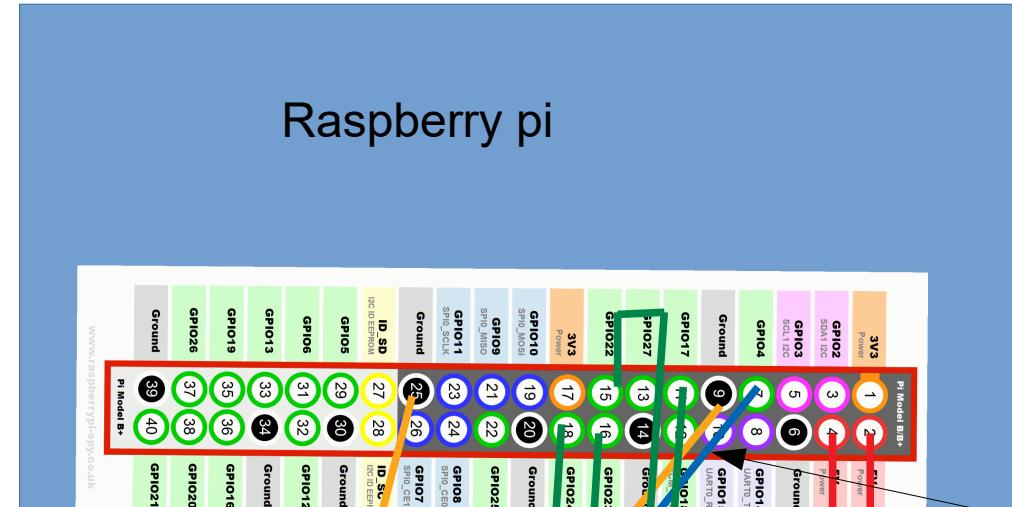
2 Le plan de cablage des composants

2.1 Pour la partie raspberry : vidéo et éclairage pièce principale

Il s'agit d'une option de commande du relais de la lumière par le raspberry. Je n'ai pas de camera Infra-Rouge, et cela permet de voir zn période nocturne et de simuler une présence. Cette option est présentée dans les photos et schémas ci-dessous.

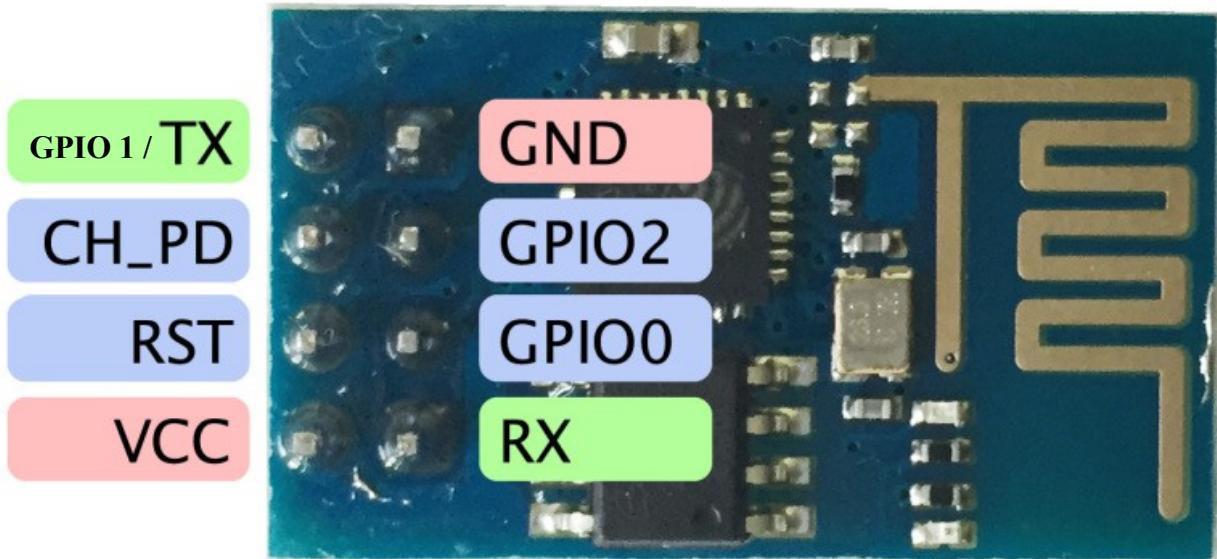
L'autre option est de commander la lumière à travers un dispositif distant WIFI à base de ESP8266-01, présenté aussi dans ce document pour lancer un arrosage jardin. Le mécanisme st le même, le code ESP-8266 est identique à un mot près (arrosage.cam vs lumiere.cam) . Voir plus loin dans ce document.

L'archive fournie est capable de faire fonctionner les 2 options sans modification de code.



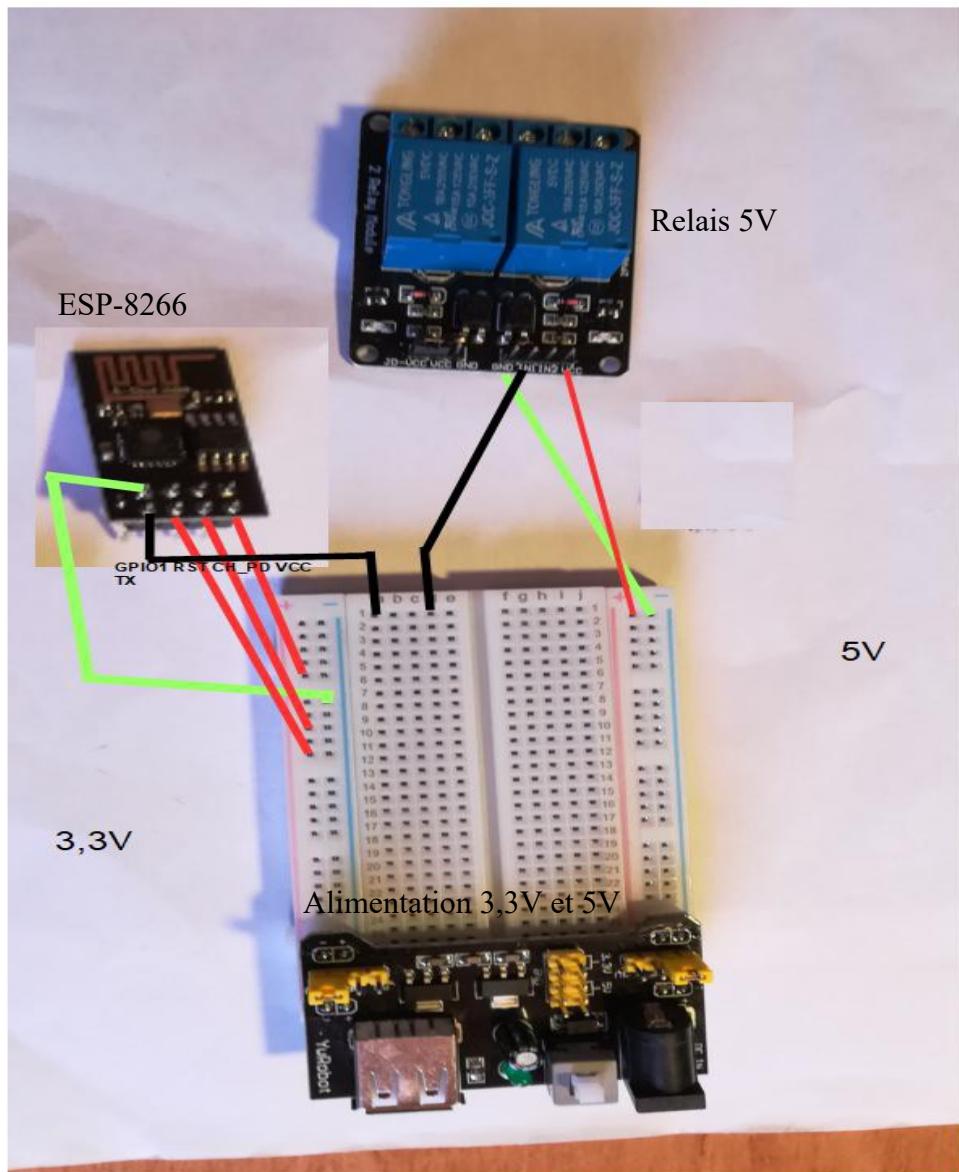
2.2 Partie distante esp-8266-01

- Dans le cas de l'option 1, pour arrosage simplement
- pour l'option 2 arrosage et éclairage. Il faudra bien identifier les 2 systèmes en fonction du flashage de l'ESP-8266

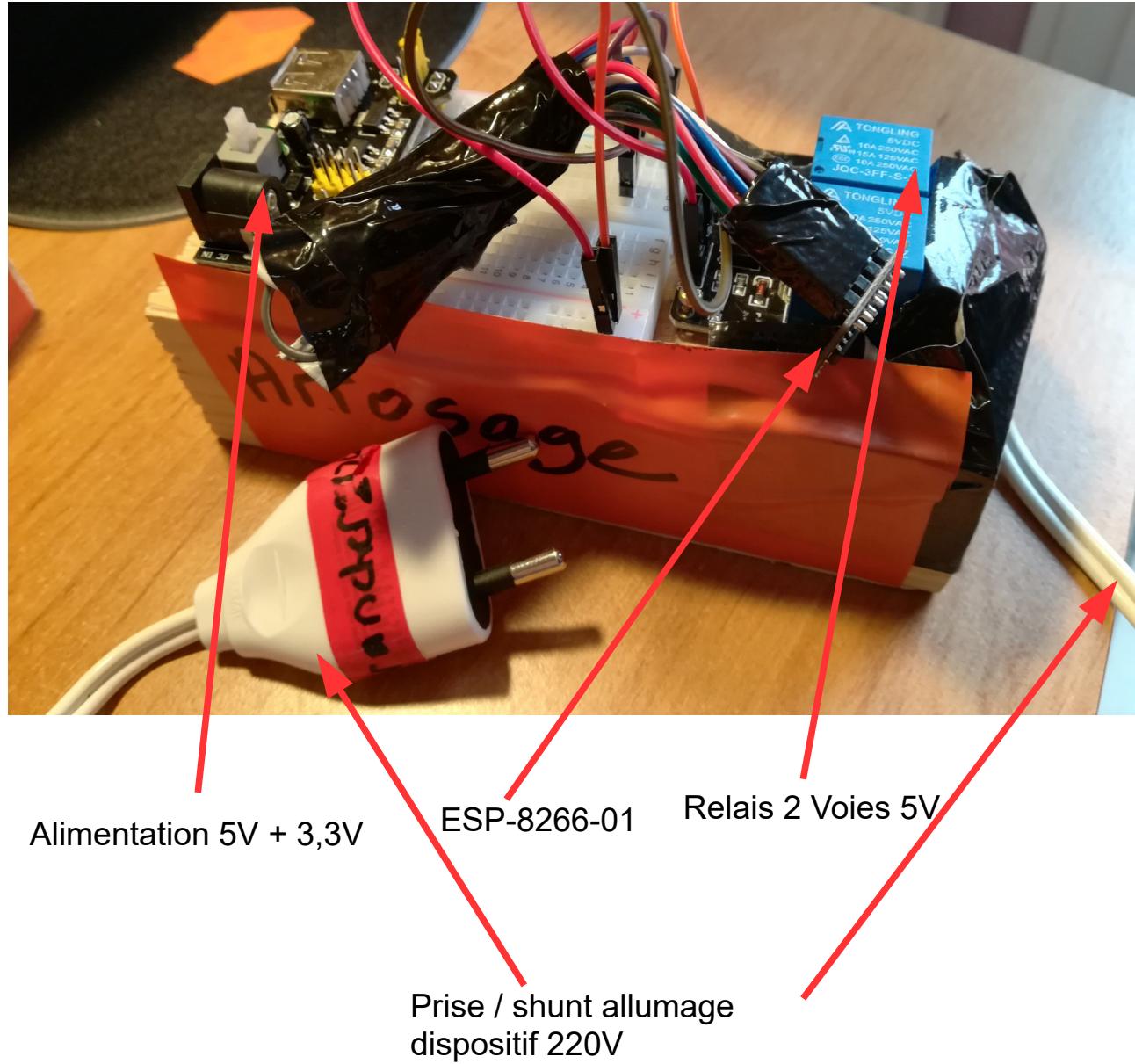


- Pour VCC et GND, il est nécessaire d'utiliser une alimentation 3.3 V externe, car l'ESP-01 peut être très gourmand (~300 mA), particulièrement lorsque le WiFi se met en route. J'ai utilisé une alimentation fournissant le 5 V (pour le relais) et 3,3V pour l'ESP-8266-01. Le niveau haut de GPIO 1 suffit pour faire tirer le relais, en alimentant ce dernier en 5V
- La broche CH_PD (*chip power-down*) doit être obligatoirement mise à VCC pour que le module soit actif.
- Optionnellement, on peut aussi mettre RST à VCC pour s'assurer qu'il n'y a pas de *reset* intempestifs. Ou connecter RST avec une pull-up de 10 kΩ à VCC et un bouton poussoir à GND.
- Dans notre cas pour actionner les relais on utilise la borne GPIO1/TX.

Ci-dessous le schéma plan de cablage de L'ES P-82666 avec son alimentation et son relais :



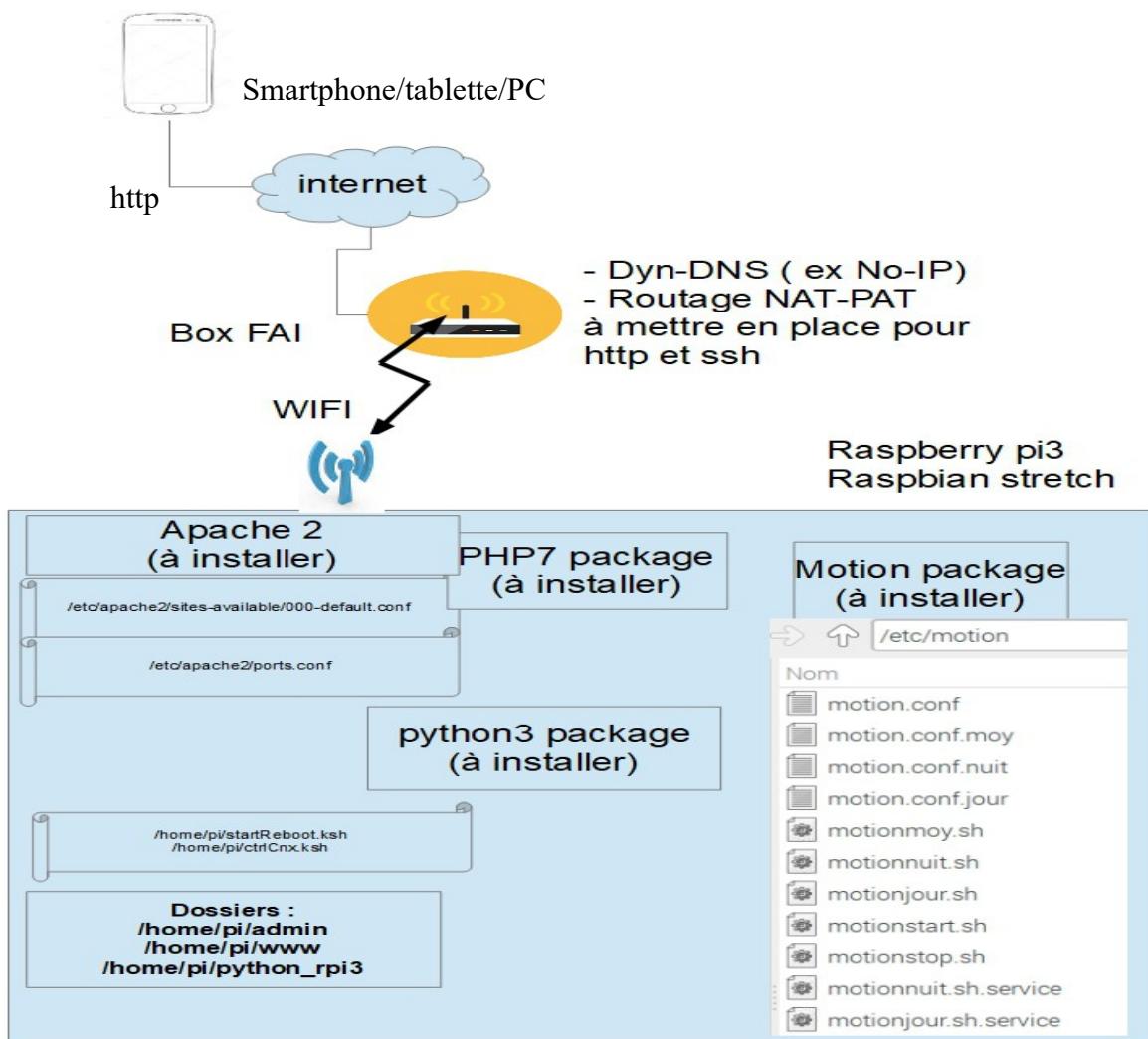
Et une photo de la réalisation :



3 Mise en place du serveur Web sur le Raspberry 3

On suppose que la partie configuration de la box et fourniture de l'accès internet via la box est réalisée, et que notre application sera joignable depuis internet à l'aide d'une URL (si c'est no-ip) du type **http://<sous-domaine>.hopto.org:<port>**/ si vous avez redirigé le <port> vers le port 80 de votre raspberry pi (80 est le port d'écoute de Apache sur votre raspberry pi). <port> doit être inférieur à **65535** . Le sous-domaine est donné par **no-ip** lors de la création de votre compte DNS. Voir les liens définis dans le paragraphe des pré-requis ([§ 1.1](#))

L'application repose sur l'architecture technique et logicielle schématisée ci-dessous :



Les pré-requis sont l'installation des packages Apache2 ,Motion, PHP7 et Python3 .

Les vérifications de la présences du package se font dans une fenêtre de commande (terminal) :

```
dpkg -l | grep -i apache2  
dpkg -l | grep -i motion  
dpkg -l | grep -i php  
dpkg -l | grep -i python3
```

Si vous n'obtenez pas des réponses sur l'un des package, il faut l'installer avec **apt-get** à choisir donc dans la liste ci-dessous :

```
sudo apt-get install apache2  
sudo apt-get install motion  
sudo apt-get install php  
sudo apt-get install python3
```

Les dossiers et fichiers décrits dans le schéma sont livrés dans l'archive, il conviendra ensuite de les positionner dans les répertoires identifiés dans le schéma ci-dessus et de leur donner le bon propriétaire et les bons droits. Voir le readme de l'archive [github](#)

4 Mise en place des scripts Raspberry

Ces scripts sont fournis dans l'archive [github](#) .

On suppose que la mise en place de Raspbian a été faîte, que les packages apache2, motion, php, python3, évoqués ci-dessus, ont été installés, que le wifi est activé sur le raspberry et que celui ci est accessible depuis le réseau local à travers la box.

Pour transférer le zip sur votre raspberry deux façons :

- depuis l'interface graphique de votre raspberry (à travers RealVNC), lancer le navigateur internet et récupérer le projet zip sur [github](#).
- ou installer un serveur ftp sur le raspberry (par exemple [vsftpd](#)) et configurer pour le lancer en standalone en cas de besoin de transfert et installer un client Filezilla sur votre PC.

Il suffit de décompresser l'archive récupérée de [github](#) sous le répertoire **/home/pi**. Ensuite on suit la procédure suivante :

- Les 2 fichiers **/home/pi/startReboot.ksh** et **/home/pi/ctrlCnx.ksh** doivent avoir le propriétaire pi et le droit d'exécution x pour le user pi le groupe pi

- le répertoire **/home/pi/www** doit être au propriétaire pi et doit avoir les droits **rwxr-xr-x**
- le contenu de **/home/pi/www (html et html2)** doit être au propriétaire pi et tous les fichiers de types **.py .php** doivent avoir le droit d'exécution **x** pour le user pi le groupe pi .
- Le répertoire **/home/pi/python_rpi3** et le contenu doit être au propriétaire pi et doit avoir les droits **rwxr-xr-x** . On trouvera dans ce répertoire le fichier **rotcamera.service** pour la mise en place de ce service évoqué dans [ce paragraphe](#).
- Le répertoire **/home/pi/Dropbox-Uploader** et le contenu doit être au propriétaire pi et poursuivre par la mise en place de Dropbox comme indiqué dans [ce paragraphe](#).
- Sous compte **root** (**sudo su**) , aller dans le répertoire **/home/pi/etc_apache2** , faire une copie récursive du contenu du répertoire dans **/etc/apache2** :

– **cp -R /home/pi/etc_apache2 /etc/apache2**

Attention pas d'espace entre / et home !!!

- Sous compte **root** (**sudo su**) , aller dans le répertoire **/home/pi/etc_systemd_system** , faire une copie récursive du contenu du répertoire dans **/etc/systemd/system** :

– **cp -R /home/pi/etc_systemd_system /etc/systemd/system**

Attention pas d'espace entre / et home !!!

Le fichier **rotcamera.service** doit être propriétaire **root**.

- On mettra en place, si on ne l'a pas fait avant, **rotcamera.service** comme évoqué dans [ce paragraphe](#).
- Sous compte **root** (**sudo su**) , aller dans le répertoire **/home/pi/etc_motion**, faire une copie récursive du contenu du répertoire dans **/etc/motion** :

– **cp -R /home/pi/home/pi/etc_motion /etc/motion**

Attention pas d'espace entre / et home !!!

- Mise en place du crontab user décrit dans [ce paragraphe](#)

Sous utilisateur pi lancer la commande : **crontab -e**

En fin de fichier rajouter les 3 lignes suivantes :

```
*/20 1-23 * * * /home/pi/ctrlCnx.ksh
0 0 * * * sudo reboot
@reboot /home/pi/startReboot.ksh
```

- Rebooter votre raspberry pi, puis tester le serveur apache en seayant de vous connecter en local : <http://raspberrypi/> . Remplacer **raspberrypi** par le nom DNS donné par votre Box.

- Si OK, tenter une connexion depuis internet depuis votre portable **en désactivant le wifi** sur celui-ci, avec l'URL donné par votre DynDNS par exemple no-ip, et votre port de forwarding configuré dans votre box (NAT-PAT), vous devriez pour no-ip avoir quelque chose comme :
http://<sous-domaine>.hopto.org:<portForwardé>/

4.1 Contrôle de la connexion wifi

Automatiquement mis en place en dézippant l'archive [github](#).

Fichier **/home/pi/ctrlCnx.ksh**

```
#!/bin/bash
ping -c 1 192.168.1.1
if [ $? -ne 0 ]; then
    sleep 1
    ping -c 1 192.168.1.1
    if [ $? -ne 0 ]; then
        sleep 1
        ping -c 1 192.168.1.1
        if [ $? -ne 0 ]; then
            sudo reboot
        fi
    fi
fi
```

Vérifie 3 fois si la Box est accessible, sinon un reboot est relancé.

Rendre exécutable :

```
chmod 755 /home/pi/ctrlCnx.ksh
```

4.2 Relance de la camera au reboot

Automatiquement mis en place en dézippant l'archive [github](#).

Fichier **/home/pi/startReboot.ksh**

```
#!/bin/bash
sleep 120
sudo /etc/motion/motionstart.sh
sleep 10
sudo /usr/sbin/apachectl restart
```

Démarre la capture de mouvement 2 mn après le reboot (laisse certains services se lancer avant), puis relance apache.

Rendre exécutable :

chmod 755 /home/pi/startReboot.ksh

4.3 Transfert automatique des photos et vidéos sur Dropbox

Automatiquement mis en place en dézippant l'archive [github](#).

Fichier **/home/pi/python_rpi3/dropbox.py**

```
#!/usr/bin/python3
import os
import time
import subprocess
if __name__ == "__main__":
    while True:
        liste=os.listdir("/home/pi/motion")
        for file in liste:
            age=0
            # st_mtime date in sec
            try:
                last_time= round(os.stat("/home/pi/motion/"+file).st_mtime )
                # time.time() current time in seconds
                age=round(time.time() -last_time)
            except:
                age=0
                pass
            if (os.path.exists("/home/pi/motion/"+file) and file.endswith("jpg")):
                cmde="/home/pi/Dropbox-Uploader/dropbox_uploader.sh upload
                /home/pi/motion/"+file+" /"
                try :
                    p = subprocess.Popen(cmde,shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE)
                    p.wait()
                    # (output, err) = p.communicate()
                except :
                    pass
                if (age > 86400) :
                    os.remove("/home/pi/motion/"+file);
            if (os.path.exists("/home/pi/motion/"+file) and file.endswith("mkv")) :
```

```
cmde="/home/pi/Dropbox-Uploader/dropbox_uploader.sh upload  
/home/pi/motion/" +file" /"  
if (age > 40 and os.stat("/home/pi/motion/" +file).st_size > 1024000) :  
    try :  
        p =  
subprocess.Popen(cmde,shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE)  
        p.wait()  
        # (output, err) = p.communicate()  
    except :  
        pass  
if (age > 86400 or (age > 40 and os.stat("/home/pi/motion/" +file).st_size < 1024000)) :  
    try :  
        os.remove("/home/pi/motion/" +file);  
    except :  
        pass  
  
time.sleep(10)
```

4.4 Mise en place d'un crontab user

Commande **crontab -e**

En fin de fichier rajouter les 3 lignes suivantes :

```
*/20 1-23 * * * /home/pi/ctrlCnx.ksh  
0 0 * * * sudo reboot  
@reboot /home/pi/startReboot.ksh
```

Bien respecter les espaces dans ce fichier.

La première ligne vérifie toutes les 20 mn , entre 1h et 23 h, que la connexion avec la livebox est correcte sinon reboot (voir plus haut)

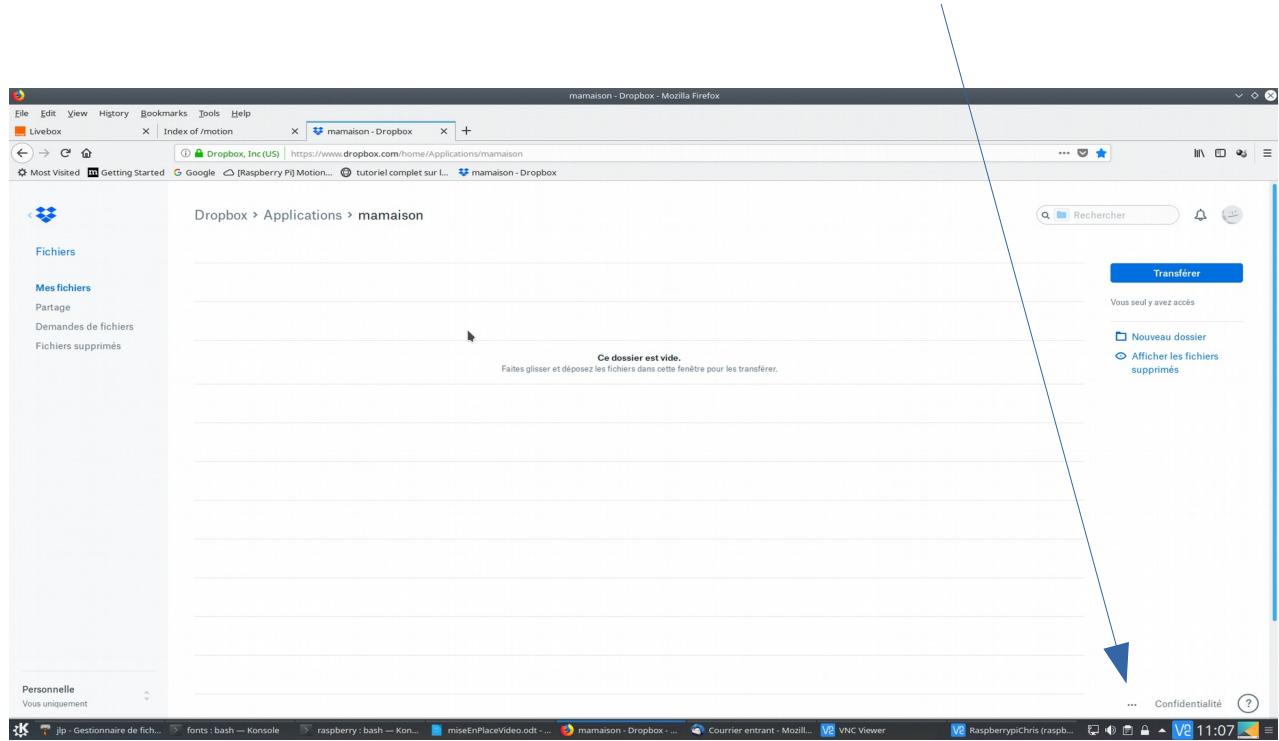
La deuxième ligne force un reboot à minuit

La 3ieme ligne, 2mn après le reboot, lance la détection de mouvement par la caméra

5 Compte Dropbox

Ouvrir un compte Dropbox sous [dropbox.com.](https://www.dropbox.com/)

Créer une application à partir des 3 points en bas de la page et menu développeur.



Puis dans le menu choisir Développeur et créer l'application **mamaison**.

Supprimer le fichier **/home/pi/.dropbox_uploader**

Puis générer un token, que l'on reporterà à la demande du premier lancement script :

/home/pi/Dropbox-Uploader/dropbox_uploader.sh

Copier ensuite le fichier **/home/pi/.dropbox_uploader** sous **/root/.dropbox_uploader** en suivant les commandes suivantes :

```
sudo su  
cd /root  
cp /home/pi/.dropbox_uploader .
```

6 Gestion camera et nettoyage vidéo

Ci dessous le script complet (relance du service dropbox qui part en sucette lors de la suppression des fichiers)

Il faut mettre en place ce programme de gestion camera et autre
(/home/pi/python_rpi3/rotCamera.py) comme un service à lancer au démarrage du rpi3. La création du service est décrite [ici](#)

En résumé pour créer le service suivre le mode opératoire ci-dessous.

Créer le fichier **/home/pi/python_rpi3/rotcamera.service** (il est dans l'archive [github](#)) ci dessous :

```
[Unit]
Description=Pilotage Camera
After=multi-user.target

[Service]
Type=simple
ExecStart=/usr/bin/python3 /home/pi/python_rpi3/rotCamera.py
Restart=on-abort

[Install]
WantedBy=multi-user.target
```

Puis :

```
sudo cp /home/pi/python_rpi3/rotcamera.service /etc/systemd/system rotcamera.service
sudo chown root:root /etc/systemd/system/rotcamera.service
sudo chmod 666 /etc/systemd/system/rotcamera.service
```

Et on active le service :

```
sudo systemctl enable rotcamera.service
```

Pour connaître son état :

```
service rotcamera status
```

exemple de retour de **service rotcamera status**

```
* rotcamera.service - Pilotage Camera
   Loaded: loaded (/lib/systemd/system/rotcamera.service; enabled; vendor preset
```

```
Active: active (running) since Mon 2019-03-04 17:07:49 CET; 42min ago
Main PID: 503 (python3)
CGroup: /system.slice/rotcamera.service
|- 503 /usr/bin/python3 /home/pi/python_rpi3/rotCamera.py
`-5050 /usr/bin/motion
```

Fichier **/home/pi/python_rpi3/rotCamera.py** : (apporté automatiquement par l'archive [github](#)

```
import os
import time
import sys
import stat
import RPi.GPIO as GPIO
import subprocess

def eraseFiles(repertoire):
    files=os.listdir(repertoire)
    for i in range(0,len(files)):
        os.remove(repertoire+'/'+files[i])

def step_4 (p):
    if p==0:
        GPIO.output(17,0)
        GPIO.output(22,0)
        GPIO.output(23,0)
        GPIO.output(24,0)

    if p==1:
        GPIO.output(17,1)
        GPIO.output(22,1)
        GPIO.output(23,0)
        GPIO.output(24,0)

    if p==2:
        GPIO.output(17,0)
        GPIO.output(22,1)
        GPIO.output(23,1)
        GPIO.output(24,0)

    if p==3:
        GPIO.output(17,0)
```

```
GPIO.output(22,0)
GPIO.output(23,1)
GPIO.output(24,1)

if p==4:
    GPIO.output(17,1)
    GPIO.output(22,0)
    GPIO.output(23,0)
    GPIO.output(24,1)

def steps_4(value):
    print (value)
    global pas
    if(value<0):
        for i in range (0,abs(value)):
            step_4(pas)
            time.sleep(0.005)
            pas+=1
            if(pas>=5):
                pas=1;

    else:
        for i in range (0,abs(value)):
            step_4(pas)
            time.sleep(0.005)
            if(pas==1):
                pas=5;
            pas-=1
        step_4(0)

def steps_10Deg(nb10deg):
    steps_4(51*nb10deg)

def mvCamera(st) :
    global counter
    if counter+st > 12:
        steps_10Deg(12-counter)
        counter=12
        fichier = open("/home/pi/www/html/counter.cam", "w")
        fichier.write("12")
        fichier.close()
    if counter+st <0 :
        steps_10Deg(-counter)
        counter=0
```

```
fichier = open("/home/pi/www/html/counter.cam", "w")
fichier.write("0")
fichier.close()
if counter+st >=0 and counter+st <=12 :
    print("4 phase moving")
    steps_10Deg(st)
    counter+=st
    fichier = open("/home/pi/www/html/counter.cam", "w")
    fichier.write(str(counter))
    fichier.close()
    print("counter=",counter)

if __name__ == "__main__":
    counter=0
    if os.path.exists("/home/pi/www/html/counter.cam"):
        fichier = open("/home/pi/www/html/counter.cam", "r")
        counter=int(fichier.readline().rstrip('\n\r'))
        fichier.close()
    else:
        fichier = open("/home/pi/www/html/counter.cam", "w")
        fichier.write("0")
        fichier.close()

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(17, GPIO.OUT)
GPIO.setup(22, GPIO.OUT)
GPIO.setup(23, GPIO.OUT)
GPIO.setup(24, GPIO.OUT)

GPIO.setwarnings(False)
# defini le port GPIO 4 comme etant une sortie output
GPIO.setup(4, GPIO.OUT)
GPIO.output(4,GPIO.HIGH)
GPIO.setup(20, GPIO.OUT)
GPIO.output(20,GPIO.HIGH)
step_4(0)
pas=1
# detection fichier gauche.cam droite.cam calgauche.cam caldroite.cam
while True :
    if os.path.exists("/home/pi/www/html/droite.cam"):
        os.remove("/home/pi/www/html/droite.cam")
        mvCamera(1)
    if os.path.exists("/home/pi/www/html/gauche.cam"):
```

```
os.remove("/home/pi/www/html/gauche.cam")
mvCamera(-1)
if os.path.exists("/home/pi/www/html/calgauche.cam"):
    os.remove("/home/pi/www/html/calgauche.cam")
    counter=0
    fichier = open("/home/pi/www/html/counter.cam", "a")
    fichier.write("0")
    fichier.close()
if os.path.exists("/home/pi/www/html/caldoite.cam"):
    os.remove("/home/pi/www/html/caldoite.cam")
    counter=12
    fichier = open("/home/pi/www/html/counter.cam", "a")
    fichier.write("12")
    fichier.close()
# ne pas eclairer ou arroser plus de une heure
if os.path.exists("/home/pi/www/html/lumiere.cam"):
    GPIO.output(4,GPIO.LOW)
    if (time.time() - os.stat("/home/pi/www/html/lumiere.cam")[stat.ST_MTIME] > 3600) :
        os.remove("/home/pi/www/html/lumiere.cam")
else :
    GPIO.output(4,GPIO.HIGH)

if os.path.exists("/home/pi/www/html/arrosage.cam"):
    GPIO.output(20,GPIO.LOW)
    if (time.time() - os.stat("/home/pi/www/html/arrosage.cam")[stat.ST_MTIME] > 3600) :
        os.remove("/home/pi/www/html/arrosage.cam")
        if os.path.exists("/home/pi/www/html2/arrosage.cam"):
            os.remove("/home/pi/www/html2/arrosage.cam")
        if os.path.exists("/home/pi/www/arrosage.cam"):
            os.remove("/home/pi/www/arrosage.cam")
    else :
        GPIO.output(20,GPIO.HIGH)

if os.path.exists("/home/pi/www/html/admin/delmovies.cam"):
    os.remove("/home/pi/www/html/admin/delmovies.cam")
    eraseFiles("/home/pi/www/html/motion")
    time.sleep(1)
    cmde="sudo systemctl restart dropbox"
    try :
        p = subprocess.Popen(cmde,shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE)
        p.wait()
        # (output, err) = p.communicate()
    except :
        pass
```

```
if os.path.exists("/home/pi/www/html/admin/startcamera.cam"):
    os.remove("/home/pi/www/html/admin/startcamera.cam")
    os.system("sudo /etc/motion/motionstart.sh")
if os.path.exists("/home/pi/www/html/admin/stopcamera.cam"):
    os.remove("/home/pi/www/html/admin/stopcamera.cam")
    os.system("sudo /etc/motion/motionstop.sh")

time.sleep(0.5)
```

7 Flashage des dispositifs ESP-8266-01

Pour cela, sur mon PC Windows 10, j'utilise [l'IDE Arduino](#), et le plugin nécessaire à la programmation [ESP-8266-01](#) => voir paraphe VI et suivant.

Les 2 fichiers de programmations pour les relais de gestion de la lumiere et de l'arrosage sont disponibles sous le répertoire **/home/pi/esp-8266-flash** (fourni dans l'archive [github](#)) :

/home/pi/esp-8266-flash/arrosageESP8266.ino

```
/***
 * BasicHTTPClient.ino
 *
 * Created on: 24.05.2015
 *
 */
#include <Arduino.h>

#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>

#include <ESP8266HTTPClient.h>

#ifndef USE_SERIAL Serial

ESP8266WiFiMulti WiFiMulti;

void setup() {
    pinMode(1, OUTPUT);
    digitalWrite(1, LOW);
    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, LOW);

    for(uint8_t t = 4; t > 0; t--) {
        // USE_SERIAL.printf("[SETUP] WAIT %d...\n", t);
        delay(1000);
    }
}
```

```
// USE_SERIAL.flush();
delay(1000);
}
// Remplacer le <nom de votre reseau> et le <mot de passe> ci-dessous :
// Nom réseau = nom de la box , mot de passe = mot de passe WPA en général 26 caractères choisis parmi
0123456789ABCDEF
WiFiMulti.addAP("<nom de votre reseau>", "<mot de passe>");

}

void loop() {
    // wait for WiFi connection
    if((WiFiMulti.run() == WL_CONNECTED)) {

        HTTPClient http;

        // Vérifier le nom DNS donné par votre box à votre raspberry ( pour moi raspberrypi) lors de la création de
        l'adresse fixe
        // ainsi que le port du virtual host 8853 à configurer dans les fichiers apache 2 :
        // - /etc/apache2/sites-available/000-default.conf ( virtualhost port 8853 mais on peut choisir un autre port autre que
        80 et 8080 )
        // - /etc/apache2/ports.conf ( Listen 8853)
        http.begin("http://raspberrypi:8853/arrosage.cam"); //HTTP

        // USE_SERIAL.print("[HTTP] GET...\n");
        // start connection and send HTTP header
        int httpCode = http.GET();

        // httpCode will be negative on error
        if(httpCode > 0) {
            // HTTP header has been sent and Server response header has been handled
            // USE_SERIAL.printf("[HTTP] GET... code: %d\n", httpCode);

            // file found at server
            if(httpCode == HTTP_CODE_OK) {
                digitalWrite(1, HIGH);
                digitalWrite(LED_BUILTIN, LOW);
                // USE_SERIAL.printf("[HTTP] GET... codeOK: %d\n", httpCode);

                //String payload = http.getString();
                // USE_SERIAL.println(payload);

            }
            else
            {
                digitalWrite(1, LOW);
                digitalWrite(LED_BUILTIN, HIGH);
            }
        }
    }
}
```

```
// USE_SERIAL.printf("[HTTP] GET... codeKO: %d\n", httpCode);

    }
} else {

    digitalWrite(1, LOW);
    digitalWrite(LED_BUILTIN, HIGH);
}

http.end();
}

delay(2000);
}
```

/home/pi/esp-8266-flash/lumiereESP8266.ino

```
/***
 * BasicHTTPClient.ino
 *
 * Created on: 24.05.2015
 *
 */

#include <Arduino.h>

#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>

#include <ESP8266HTTPClient.h>

#ifndef USE_SERIAL Serial

ESP8266WiFiMulti WiFiMulti;

void setup() {
    pinMode(1, OUTPUT);
    digitalWrite(1, LOW);
    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, LOW);

    for(uint8_t t = 4; t > 0; t--) {

        delay(1000);
    }

    // Remplacer le <nom de votre reseau> et le <mot de passe> ci-dessous :
}
```

```
// Nom réseau = nom de la box , mot de passe = mot de passe WPA en général 26 caractères choisis parmi  
0123456789ABCDEF  
WiFiMulti.addAP("<nom de votre réseau>", "<mot de passe>");  
  
}  
  
void loop() {  
    // wait for WiFi connection  
    if((WiFiMulti.run() == WL_CONNECTED)) {  
  
        HTTPClient http;  
  
        // configurer traged server and url  
  
        // Vérifier le nom DNS donné par votre box à votre raspberry ( pour moi raspberrypi) lors de la création de  
l'adresse fixe  
        // ainsi que le port du virtual host 8853 à configurer dans les fichiers apache 2 :  
        // - /etc/apache2/sites-available/000-default.conf ( virtualhost port 8853)  
        // - /etc/apache2/ports.conf ( Listen 8853 mais on peut choisir un autre port autre que 80 et 8080)  
        // Gestion d'une lampe d'éclairage  
        http.begin("http://raspberrypi:8853/lumiere.cam"); //HTTP  
  
        // start connection and send HTTP header  
        int httpCode = http.GET();  
  
        // httpCode will be negative on error  
        if(httpCode > 0) {  
            // HTTP header has been sent and Server response header has been handled  
  
            // file found at server  
            if(httpCode == HTTP_CODE_OK) {  
                digitalWrite(1, HIGH);  
                digitalWrite(LED_BUILTIN, LOW);  
  
            }  
            else  
            {  
                digitalWrite(1, LOW);  
                digitalWrite(LED_BUILTIN, HIGH);  
  
            }  
        } else {  
            digitalWrite(1, LOW);  
        }  
    }  
}
```

```

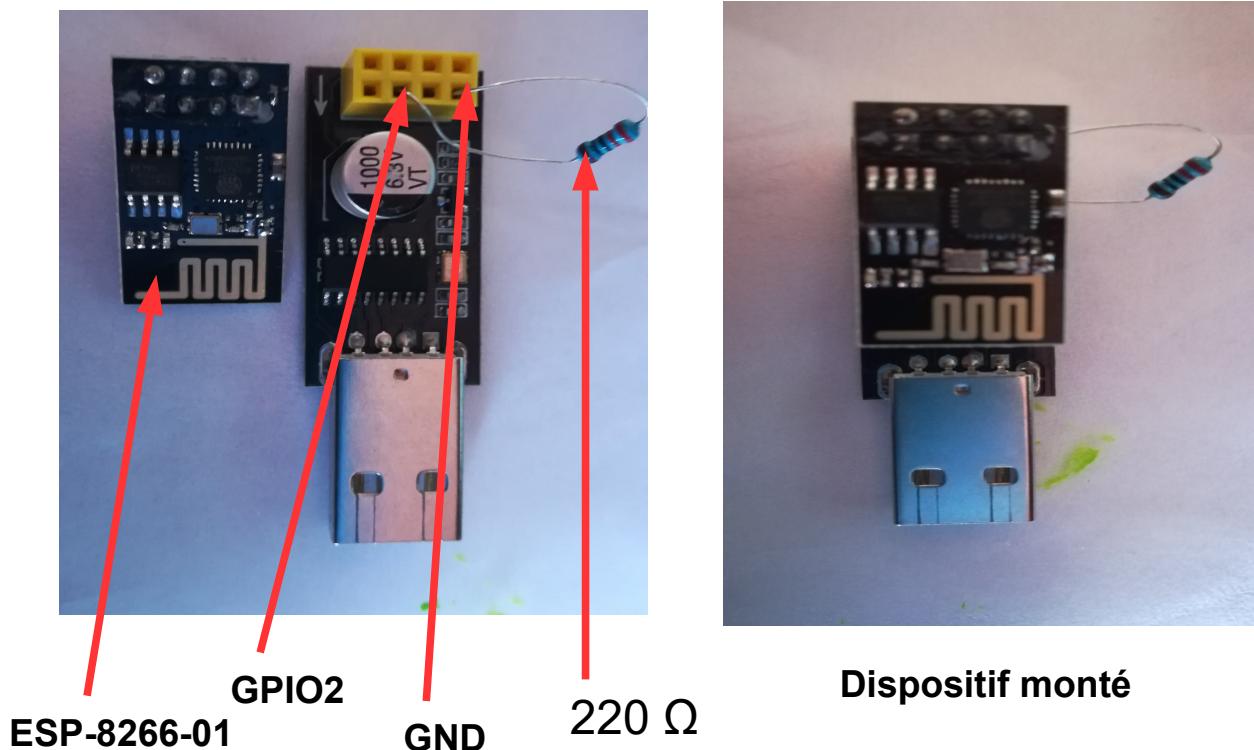
        digitalWrite(LED_BUILTIN, HIGH);
    }

    http.end();
}

delay(2000);
}

```

Pour la programmation j'utilise de dispositif de la photo suivante, et pour pouvoir téléverser les programmes, il faut mettre la sortie GPIO2 sur la Masse. On a les montages suivants :



Après avoir lancé l'IDE Arduino, on branche l'ensemble sur une sortie USB (Par la suite on veillera à utiliser la même). Il se peut que la première fois l'IDE Arduino ne détecte pas le port COM de l'ESP-8266.

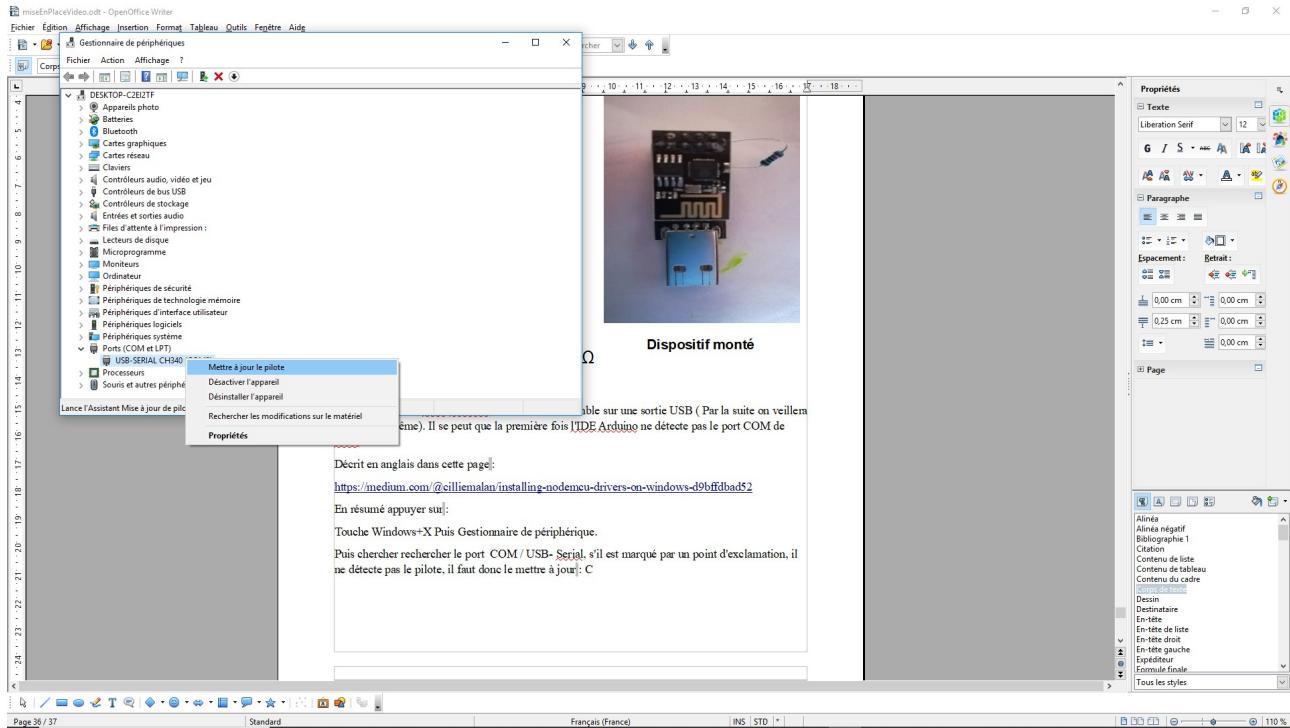
Décris en anglais dans cette page :

<https://medium.com/@cillieman/installing-nodemcu-drivers-on-windows-d9bfffdbad52>

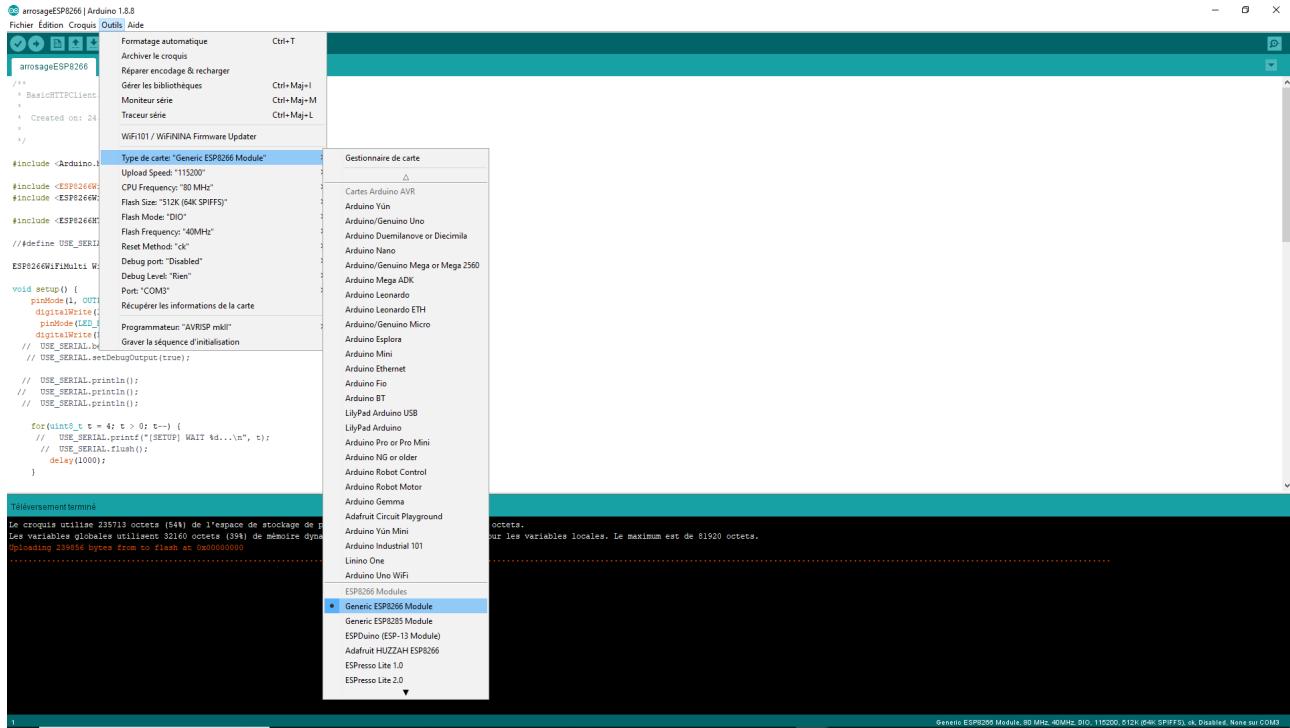
En résumé, appuyer sur :

Touche Windows+X Puis Gestionnaire de périphérique.

Puis chercher rechercher le port COM / USB- Serial, s'il est marqué par un point d'exclamation, il ne détecte pas le pilote, il faut donc le mettre à jour : Clic droit / Mettre à jour le pilote, et laisser Windows chercher le pilote.



Pour la programmation dans l'IDE Arduino choisir ESP-8266 Generic module



Après chaque téléversement, ou échec d'un téléversement, pour ce remettre en mode programmation, il faut retirer le dispositif du port USB et le rebrancher après quelques secondes (pour réinitialise GPIO2 à la masse). Une fois téléversé, le programme commence à s'exécuter.

Il faut ensuite passer à la réalisation des montages de ces systèmes distants comme indiqué dans le schéma du [ce paragraphe](#).

