

# DOCUMENTATION DU PROJET DE STAGE

“Installation et configuration d’une caméra de sécurité avec retour vidéo à distance”

I - Installation du Raspberry Pi.....	1
II - Pourquoi l’utilisation de Motion et de MotionEye ?.....	2
III - Installation de Motion et MotionEye (et des différentes dépendance).....	2
IV - Accès à Motion ou MotionEye.....	3
V - Problèmes rencontrés et solutions trouvées au cours des différentes installations.....	4
VI - Mise en place de la redirection des ports.....	5
VII - Sécurisation du Raspberry Pi et de MotionEye.....	5
VIII - Utilisation des flux vidéo sur une page externe (type html/css).....	5
IX - Achat d’un service d’hébergement web.....	6
IX - Création du site web pour visualisation des flux vidéos.....	6
X - Mise en place d’un capteur de température.....	6
XI - Installation et mise en place de l’utilisation d’un router.....	7

## I - Installation du Raspberry Pi

J’installe le système d’exploitation “RASPBERY PI OS (32-BIT)” via le logiciel officiel: Raspberry Pi Imager v1.7.4, directement sur ma carte SD.

☒ Norm d'hôte pi.local

☒ Activer SSH

☒ Utiliser un mot de passe pour l'authentification

☐ Authentification via clef publique

Définir authorized\_keys pour 'raspberrypi': \_\_\_\_\_

☒ Définir nom d'utilisateur et mot de passe

Nom d'utilisateur : raspberrypi

Mot de passe : XXXXXXXXXXXXXXXXXXXX

☐ Configurer le Wi-Fi

SSID : \_\_\_\_\_

☐ SSID caché

Mot de passe : \_\_\_\_\_

☒ Afficher le mot de passe

Pays Wi-Fi : GB

☒ Définir les réglages locaux

Fuseau horaire : Europe/Paris

Type de clavier : fr

Configuration simple ayant pour mot de passe: XXXX

([Téléchargement du logiciel](#))

## II - Pourquoi l'utilisation de Motion et de MotionEye ?

Motion est un logiciel de détection de mouvement open source permettant l'utilisation de caméra et l'enregistrement des vidéos/photos en réponse aux mouvements détectés.

MotionEye est l'interface web qui permet une configuration et une utilisation avancée de manière simplifiée, il permet aussi un retour vidéo plus simple d'utilisation avec des paramètres de configuration supplémentaires.

Motion est donc obligatoire puisque MotionEye en dépend.

Ils me permettront un retour vidéo (et une configuration) directement via une page web.

## III - Installation de Motion et MotionEye (et des différentes dépendance)

Avant de commencer l'installation, je tiens à noter que mon installation de Motion Eye est faite sur un OS 32-BIT et un Raspbian 11 Bullseye (et que l'installation se fait uniquement avec python v2.7).

Les opérations nécessitent d'être en root donc d'utiliser *sudo* ou *sudo -i*.

La procédure à suivre pour une installation rapide est celle-ci:

Installation de "ffmpeg" et d'autres dépendances de motion:

```
sudo apt-get install ffmpeg libmariadb3 libpq5 libmicrohttpd12 -y
```

Installation de motion (version 32-BIT):

```
sudo wget
```

```
https://github.com/Motion-Project/motion/releases/download/release-4.3.2/pi\_buster\_motion\_4.3.2-1\_armhf.deb
```

```
sudo dpkg -i pi_buster_motion_4.3.2-1_armhf.deb
```

On désactive Motion pour pouvoir utiliser MotionEye:

```
sudo systemctl stop motion
```

```
sudo systemctl disable motion
```

Installation des dépendances à partir des différents dépôt:

```
sudo apt-get install python2 python-dev-is-python2 -y
```

```
sudo curl https://bootstrap.pypa.io/pip/2.7/get-pip.py --output get-pip.py
```

```
sudo python2 get-pip.py
```

```
sudo apt-get install libssl-dev libcurl4-openssl-dev libjpeg-dev zlib1g-dev -y
```

Installation de MotionEye (via python 2.7), il télécharge automatiquement les autres dépendances de python.

```
sudo pip2 install motioneye
```

Préparation des fichiers de configuration:

```
sudo mkdir -p /etc/motioneye
```

```
sudo cp /usr/local/share/motioneye/extra/motioneye.conf.sample  
/etc/motioneye/motioneye.conf
```

Préparation du fichier des médias:

```
sudo mkdir -p /var/lib/motioneye
```

Ajout d'un script en init pour lancer MotionEye au démarrage:

```
sudo cp /usr/local/share/motioneye/extra/motioneye.systemd-unit-local  
/etc/systemd/system/motioneye.service
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable motioneye
```

```
sudo systemctl start motioneye
```

(Liens officiels: [Informations pour l'installation](#) // [Tutoriel d'installation sur le Rasbian 11 Bullseye](#))

## IV - Accès à Motion ou MotionEye

### Avant le chapitre VI

Le premier accès à la page web de MotionEye se fait via: localhost:8765

L'accès à motion (inutile si on utilise MotionEye): localhost:8080 (ou 8081/8082 pour le flux vidéo de la caméra uniquement)

### Après le chapitre VI

L'accès à la page web de MotionEye se fait désormais via l'adresse: <http://86.233.120.69:8765/> qui est forwarder via un router.

## V - Problèmes rencontrés et solutions trouvées au cours des différentes installations

Problème: Lors de mon utilisation de Motion sur le raspberry mis à neuf tout fonctionnait bien mais après être revenu de la pause de midi (et en ayant reboot le système) Motion semble ne plus fonctionner.

Quand je le lance (*sudo systemctl start/restart motion*) je vérifie directement (via la commande *status*) et il m'affiche bien *Active (running)* mais quelques secondes après le programme passe en *deactivating(stop-sigterm)* puis en *inactive(dead)*.

Solution: Le fichier copié collé que j'avais créé pour lancer motion de manière automatique au démarrage du raspberry pi avait été corrompu, aucun moyen de trouver le problème via les journaux/logs. J'ai fini par trouver en fouillant les fichiers de configuration et en retournant dans le fichiers modifié et en le modifiant.

Problème: Lors de ma première tentative d'utilisation de MotionEye, en utilisant la commande *status* (*sudo systemctl status MotionEye*) je me rend compte que le programme est "failed" (exit-code) et qu'ils y a plusieurs messages d'erreur (import settings, import babel, import "main"...)

Solution: Il semblerait que motion ne doit pas être installée avec python3 (pip3). Mon installation via python3 a causé des problèmes de compatibilité et de corruption de fichier de configuration de motion (obligatoire pour motionEye). Il semblerait aussi que la version de Motion que je possède doit être utilisée sur un OS 64-BIT et non un OS 32-BIT (*uname -m*) armv7l. La solution serait donc de refaire une installation au propre (de 0).

Problème: Lors de ma première tentative de création d'un certificat de sécurité SSL Let's Encrypt j'ai installé certbot, ufw ainsi que nginx je me suis rendu compte que mon port forwarding ne fonctionnait plus (certainement à cause de ufw (uncomplicated firewall) )

Solution: La désinstallation des 3 packages via la commande (*sudo apt purge --auto-remove nompackage*) m'a permis de régler le problème.

Problème: Tentative d'installation d'un système de ventilation sur le raspberry pi, il ne s'agit pas du bon modèle et des radiateurs sont déjà installés sur le raspberry, empêchant d'installer correctement la ventilation.

Solution: Changement du modèle de ventilation de raspberry pi 4 vers un raspberry pi 3 model B

Problème: J'utilise un certificat SSL Let's Encrypt sur le site "cameramarennnes.fr" pour l'interface utilisateur, mais le flux vidéo de MotionEye transmis via des <iframe> n'est pas encrypté. Il ne s'affiche donc pas sur le site car le "mixed content" est impossible entre des sources http / https.

Solution: Il faut enlever le certificat SSL du site cameramarennnes.fr car je ne peux pas mettre de certificat SSL sur le flux vidéo.

## VI - Mise en place de la redirection des ports

Création d'une redirection des ports de l'adresse associé a mon raspberry pour une connexion à distance (port forwarding).

Je me connecte au service de la livebox (orange) via l'ip 192.168.1.1

Je me rend dans "configuration avancée" -> "NAT/PAT" et je créer la configuration suivante:

Applications	Port Interne	Port Externe	Protocole	Appareil	Actif
MotionEye	8765	8765	TCP	192.168.1.28	Oui
MotionEye_cam1	8081	8081	TCP/UDP	192.168.1.28	Oui
MotionEye_cam2	8082	8082	TCP/UDP	192.168.1.28	Oui

Je redémarre la box pour prendre en compte la configuration et enfin pouvoir me connecter à distance via l'adresse: 86.233.120.129:8765 via mon téléphone ou mon ordinateur portable.

## Après le chapitre XI

Changement de la redirection des ports sur la nouvelle adresse ip 192.168.1.39.

Je n'utilise plus la connexion ethernet de la box mais un router tp-link que j'ai configuré (via l'adresse: 192.168.0.1 / tplinkwifi.net ) en mode pont utilisé en wi-fi sur le raspberry pi. (la redirection des ports est quasiment identique à celle précédemment effectué)

La nouvelle adresse est donc 86.233.120.69:8765.

## VII - Sécurisation du Raspberry Pi et de MotionEye

Modification de certains mots de passes

- Le Raspberry Pi: XXXX
- Le compte administrateur de MotionEye: XXXX
- Le compte utilisateur de MotionEye: XXXX

Installation de fail2ban et configuration de celui-ci.

## VIII - Utilisation des flux vidéo sur une page externe (type html/css)

Flux vidéo en direct depuis une page web externe pour les futurs utilisateurs.

Pour faire cela j'utilise la balise html <iframe> </iframe> qui me permet d'introduire les adresses ip précédemment "forwardé" ainsi que la taille du retour vidéo:

```
<iframe src="http://x.x.x.x:xxxx/" width="320" height="240"></iframe>
```

## **IIIX - Achat d'un service d'hébergement web**

Achat du service d'hébergement web sur o2switch afin de créer une page web et la publier sur internet via le nom de domaine "cameramarennnes.fr"

## **IX - Création du site web pour visualisation des flux vidéos**

Création et utilisation du nom de domaine "cameramarennnes.fr"

Création de la page web uniquement en HMTL/CSS, site simple en one page responsive affichant les flux vidéos en direct ainsi qu'un mode clair/sombre automatique.

Création d'un Certificat SSL Let's Encrypt ainsi qu'un fichier de redirection http vers https automatique.

## **X - Mise en place d'un capteur de température**

Lors de mes premiers test avec la boîte fermée pendant plusieurs heures j'ai remarqué un gros problème de température bien trop élevé, j'ai donc installé un capteur de température constant et mis l'injecteur POE à l'extérieur de la boîte.

L'installation d'un capteur de température (AZ-Delivery DHT22 Temperature Sensor Module) directement sur les pins du raspberry pi ainsi que l'installation logiciel:

```
sudo python3 -m pip install --upgrade pip setuptools wheel
```

```
sudo pip3 install Adafruit_DHT
```

Puis je créer le script en python3 via nano:

```
sudo nano DHT22.py
```

```

import Adafruit_DHT
from time import sleep
sensor = Adafruit_DHT.DHT22
# DHT22 sensor connected to GPIO12.
pin = 12
print("[press ctrl+c to end the script]")
try: # Main program loop
    while True:
        humidity, temperature = Adafruit_DHT.read_retry(sensor,
                                                         pin)

        sleep(2.5)
        if humidity is not None and temperature is not None:
            print("Temp={0:0.1f}*C  Humidity={1:0.1f}%"
                  .format(temperature, humidity))
        else:
            print("Failed to get reading. Try again!")

# Scavenging work after the end of the program
except KeyboardInterrupt:
    print("Script end!")

```

et je le lance avec la commande suivante:

```
sudo python3 DHT22.py
```

## XI - Installation et mise en place de l'utilisation d'un router

Je vais avoir besoin d'un router tp-link que je configure en mode pont me permettant d'utiliser le wi-fi pour la raspberry pi.

Pour la configuration de celui-ci je vais sur tplinkwifi.net ou 192.168.0.1 puis je me connecte avec le compte administrateur. (mdp: XXXX)

Je fais la quick install simplifié par le router pour mettre activé le mode pont.