INSA | INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

Manuel d'utilisation :

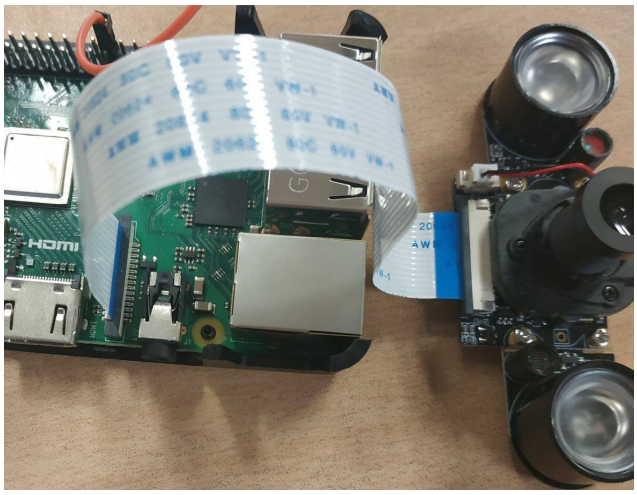
Caméra de Surveillance

Sommaire

I – Préparation de la Raspberry Pi

1) Connexion de la caméra :

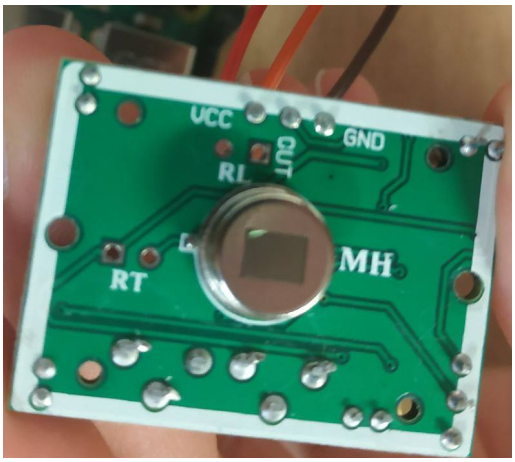
Branchez chaque broche du câble de la caméra dans le port Camera de la Raspberry Pi.

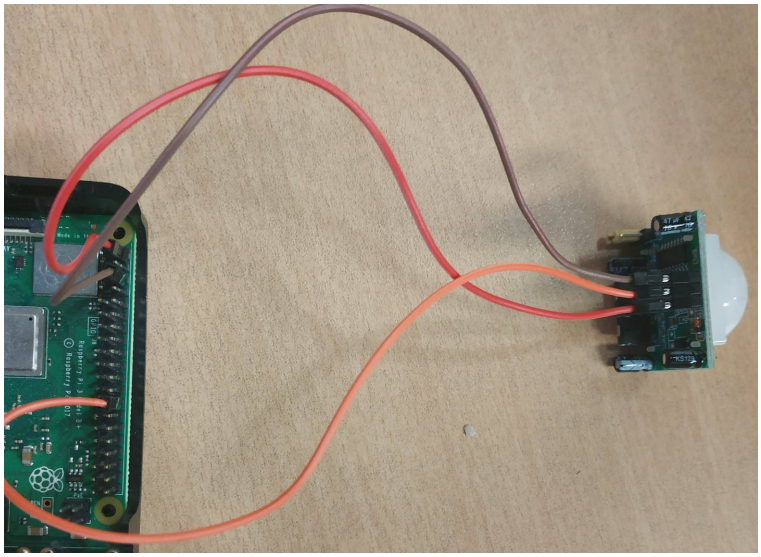


2) Connexion du PIR HC–SR501 :

Branchez :

- le pin VCC sur le pin 2 de la Raspberry Pi.
- le pin GRND sur le pin 6 de la Raspberry Pi.
- le pin OUT sur le pin 26 (GPIO 7) de la Raspberry Pi.



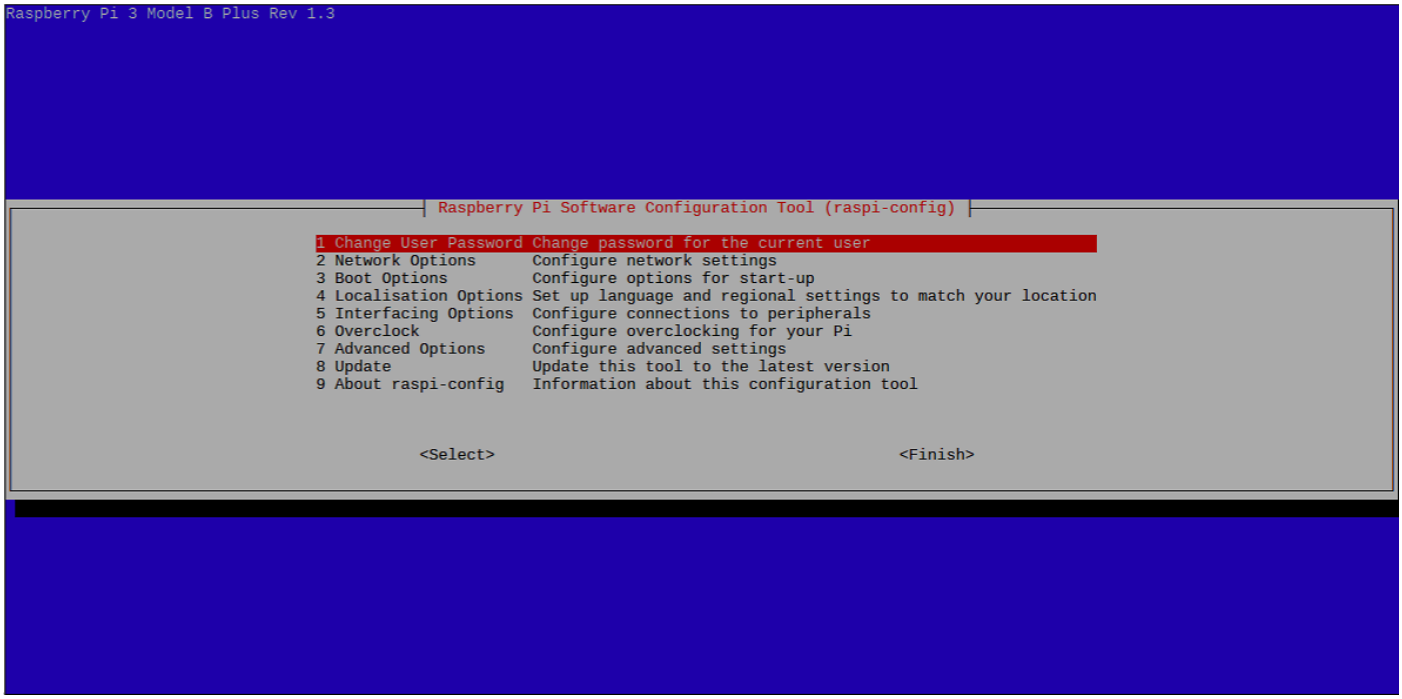


II – Configuration de la Raspberry Pi

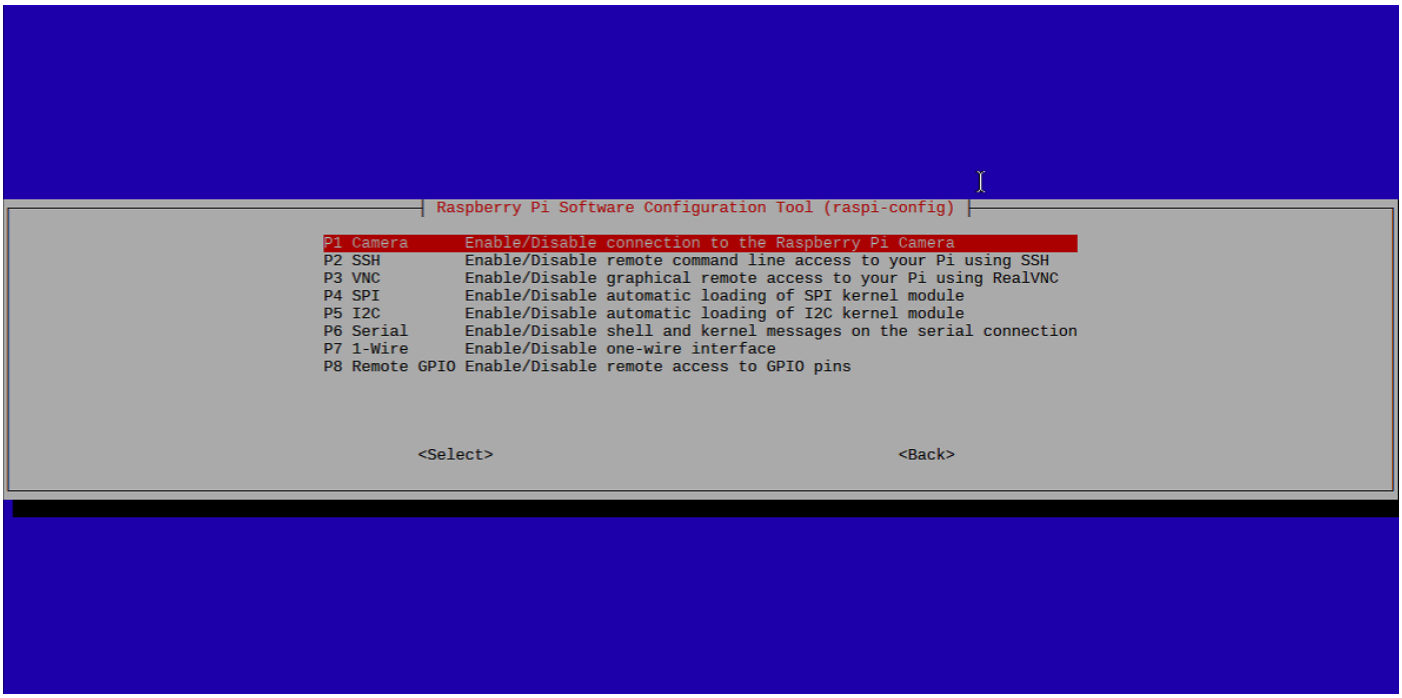
Pour activer la caméra, ouvrez le terminal et exécutez :

```
pi@raspberrypi:~ $ sudo raspi-config
```

Sélectionnez *Interface Option* :



Puis *Camera* :



Et activez-la et cliquez sur *Finish*.

III – Installation des dépendances

1) Installation OpenCV4 :

Installation des dépendances d'OpenCV4 :

Commencez par faire les mises à jour de la Raspberry Pi

```
sudo apt-get update && sudo apt-get upgrade
```

Installez les outils de développement (CMake)

```
sudo apt-get install build-essential cmake unzip pkg-config
```

Installez des bibliothèques d'images et de vidéos

```
sudo apt-get install libjpeg-dev libpng-dev libtiff-dev
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
sudo apt-get install libxvidcore-dev libx264-dev
```

Installez une interface graphique utilisateur backend (GTK)

```
sudo apt-get install libgtk-3-dev
```

```
sudo apt-get install libcanberra-gtk*
```

Installez 2 paquets

```
sudo apt-get install libatlas-base-dev gfortran
```

Installez Python 3

```
sudo apt-get install python3-dev
```

Téléchargement d'OpenCV4 pour Raspberry Pi :

Naviguez jusqu'au dossier *Home* de votre Raspberry Pi

```
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.0.0.zip
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.0.0.zip
unzip opencv.zip
unzip opencv_contrib.zip
```

Renommez les répertoires

```
mv opencv-4.0.0 opencv
mv opencv_contrib-4.0.0 opencv_contrib
```

Configuration de l'environnement virtuel Python 3 pour OpenCV4 :

Installez pip

```
wget https://bootstrap.pypa.io/get-pip.py
sudo python3 get-pip.py
```

Installez *virtualenv* et *virtualenvwrapper* pour un environnement virtuel Python

```
sudo pip install virtualenv virtualenvwrapper
sudo rm -rf ~/get-pip.py ~/.cache/pip
```

Modifiez *~/.profile*

```
echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile
echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.profile
echo "export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3" >> ~/.profile
echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile
```

Créez l'environnement virtuel OpenCV4 + Python 3

```
mkvirtualenv cv -p python3
```

Vérifons que nous somme bien dans l'environnement *cv* en tapant *workon cv*

Installez le paquet Python prérequis pour OpenCV, NumPy

```
pip install numpy
```

CMake et compilation d'OpenCV4 :

```
cd ~/opencv
mkdir build
cd build
```

Exécution CMake pour OpenCV4

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
      -D CMAKE_INSTALL_PREFIX=/usr/local \
      -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
      -D ENABLE_NEON=ON \
      -D ENABLE_VFPV3=ON \
      -D BUILD_TESTS=OFF \
      -D OPENCV_ENABLE_NONFREE=ON \
      -D INSTALL_PYTHON_EXAMPLES=OFF \
      -D BUILD_EXAMPLES=OFF ..
```

Vérifiez que dans le terminal vous avez *Non−free algorithms : YES*.

Dans *Pyhton 3* :

Interpreter : …/.virtualenvs/cv/…

Numpy : …/.virtualenvs/cv/…

Compilez OpenCV4 en tapant *make −j4*.

Si des erreurs apparaissent vous pouvez suelement faire *make*.

Installez OpenCV4

```
sudo make install
sudo ldconfig
```

Relions OpenCV4 à votre environnement virtuel Python 3 :

Créez un lien symbolique depuis l'installation d'OpenCV dans le répertoire system site−packages vers notre environnement virtuel

```
cd ~/.virtualenvs/cv/lib/python3.5/site-packages/
ln -s /usr/local/python/cv2/python-3.5/cv2.cpython-35m-arm-linux-gnueabihf.so cv2.so
```

Lancement d'OpenCV4 :

```
pi@raspberrypi:~ $ source ~/.profile
pi@raspberrypi:~ $ workon cv
(cv) pi@raspberrypi:~ $
```

2) Installation Flask :

```
pi@raspberrypi:~ $ pip install flask
```

3) Installation PiCamera :

```
pi@raspberrypi:~ $ pip install picamera
```

4) Installation Imutils :

```
pi@raspberrypi:~ $ pip install imutils
```

5) Installation Flask−BasicAuth :

```
pi@raspberrypi:~ $ pip install Flas-BasicAuth
```

IV – Personnalisation du programme

Dans le main, vous pouvez modifier les informations d'authentification au serveur de la Raspberry Pi :

```
email_update_interval = 10 #Interval de temps d'envoie d'un mail
video_camera = VideoCamera(flip=True) #Crée un objet caméra, retournez verticalement
object_classifier = cv2.CascadeClassifier("models/facial_recognition_model.xml") #OpenCV Classifier (reconnaissance facial)
```

Dans le main, la modification des paramètres :

```
app = Flask(__name__) #Appelle la dernière fonction pour lancer le serveur
app.config['BASIC_AUTH_USERNAME'] = 'admin' #Username pour accéder au site
app.config['BASIC_AUTH_PASSWORD'] = 'admin' #MDP pour accéder au site
app.config['BASIC_AUTH_FORCE'] = True
```

Vous pouvez aussi utiliser d'autre object de détection en changant *"models/facial_recognition_model.xml"* dans *object_classifier = cv2.CascadeClassifier("models/facial_recognition_model.xml")* par les autres objets de détection qui sont dans le dossier *models*.

Modifiez aussi les informations du mail :

```
fromEmail = 'adressemaile@gmail.com' #Adresse mail de l'expéditeur
fromEmailPassword = 'mdp' #MDP du compte de l'expéditeur
toEmail = 'adressemaild@gmail.com' #Adresse mail du destinataire
```

V – Lancement du programme

Pour lancer le programme, exécutez la commande dans l'environnement virtuel Python 3, OpenCV4 :

```
pi@raspberrypi:~ $ python main.py
```

Vous pouvez choisir quel type de reconnaissance vous souhaitez :

- *main.py* : reconnaissance avec la caméra (OpenCV) et envoie d'un mail avec prise d'une photo du mouvement
- *main2.py* : reconnaissance avec le PIR et envoie d'un mail avec texte
- *main2.1.py* : reconnaissance avec le PIR et envoie d'un mail et prise d'une photo du mouvement
- *main3.py* : reconnaissance avec le PIR et la caméra (facial) et envoie d'un mail et prise d'une photo du mouvement

On peut voir le flux vidéo en direct en allant sur *http://<ip_raspberry>:5000* dans un navigateur internet sur le réseau local.

Pour visionner depuis l'extérieur, vous pouvez ouvrir un port de votre box pour la Raspberry : *http://<ip_box>:<port_ouvert>* (Méthode non sécurisée).

De plus, pour un fonctionnement optimal, vous pouvez utiliser une IP statique pour la Raspberry pour éviter de rechercher l'IP à chaque fois.

Pour connaître l'adresse ip de la Raspberry Pi vous pouvez taper dans le terminal : *hostname −I*.

VI – Crédit

Vous pouvez vous rendre sur notre site internet : https://alexthegeek.github.io/Projet-Camera-Surveillance pour découvrir les réalisations supplémentaires faites sur le projet et découvrir les documents pour le projet.

Projet 1ère année STPI, INSA Centre Val de Loire. 2019.