Utilisation de la caméra sur le RaspberryPi3



Présentation

L’objectif est de comprendre comment on peut utiliser une caméra et ses images pour pouvoir ensuite les traiter et les exploiter.

En effet, les caméras sont maintenant intégrées dans un très grand nombre de produits.

Comme dans les caméras de vidéosurveillance, votre smartphone, aide à la marche arrière en automobile, gestion du positionnement dans l’espace sur un robot aspirateur, etc…

Il est possible de connecter (connecteur CSI) directement au Raspberry Pi une caméra spécifique, soit couleur, soit avec filtre infra-rouge.

Description des paramètres de la caméra V1.2 avec filtre IR :

Idéale pour prendre des clichés infrarouges, photographier dans l'obscurité ou en condition de faible luminosité.

Basé sur le même capteur Sony IMX219 et dotée de la technologie OmniBSI, la caméra infrarouge supporte jusqu'à 3280 x 2464 pixels en photo et 1080p30, 720p60 ou 640x480p60/90 en vidéo.

Premier test de la caméra.

Lance une console et tape les commandes suivantes :

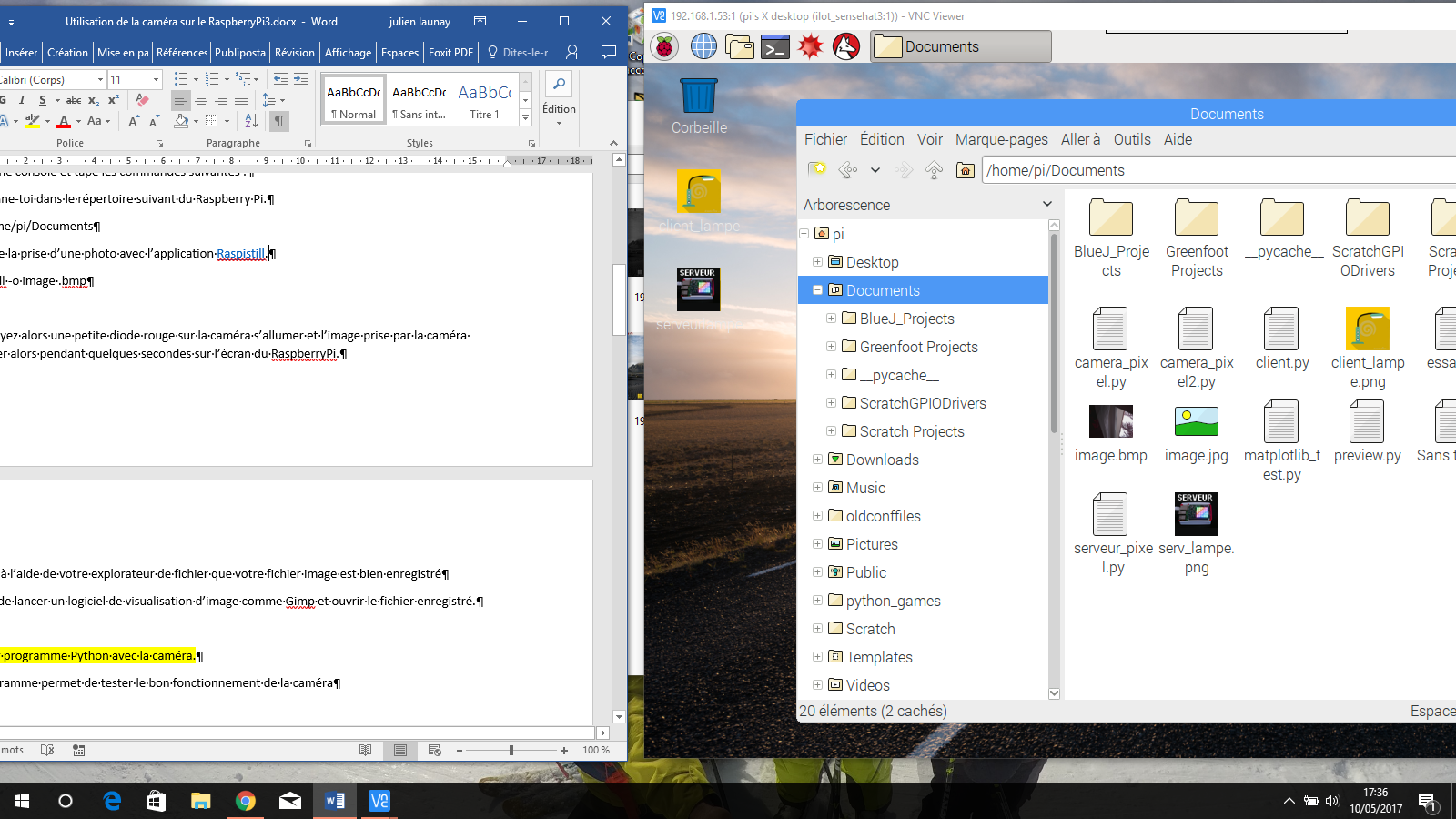
Positionne-toi dans le répertoire suivant du Raspberry Pi.

Cd /home/pi/Documents

On lance la prise d’une photo avec l’application [Raspistill](https://www.raspberrypi.org/documentation/raspbian/applications/camera.md).

Raspistill -o image .bmp

Tu vois alors une petite diode rouge sur la caméra s’allumer et l’image prise par la caméra s’afficher alors pendant quelques secondes sur l’écran du RaspberryPi.

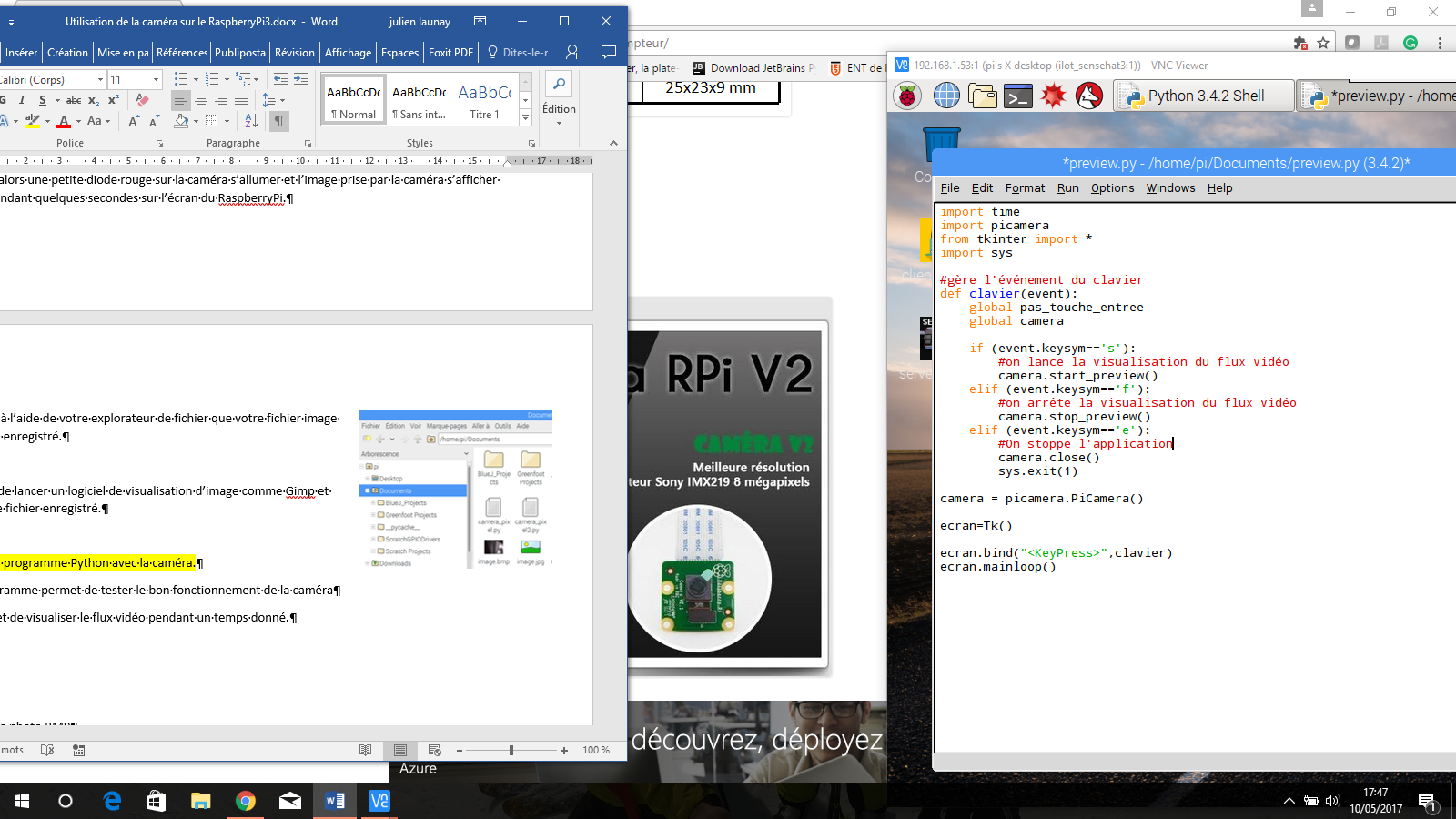
Vérifier à l’aide de votre explorateur de fichier que votre fichier image est bien enregistré.

Il suffit de lancer un logiciel de visualisation d’image comme Gimp et ouvrir le fichier enregistré.

Premier programme Python avec la caméra.

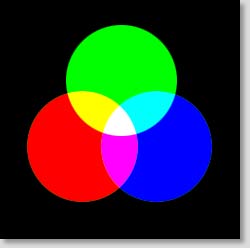
Ce programme permet de tester le bon fonctionnement de la caméra

Il permet de visualiser le flux vidéo pendant un temps donné.



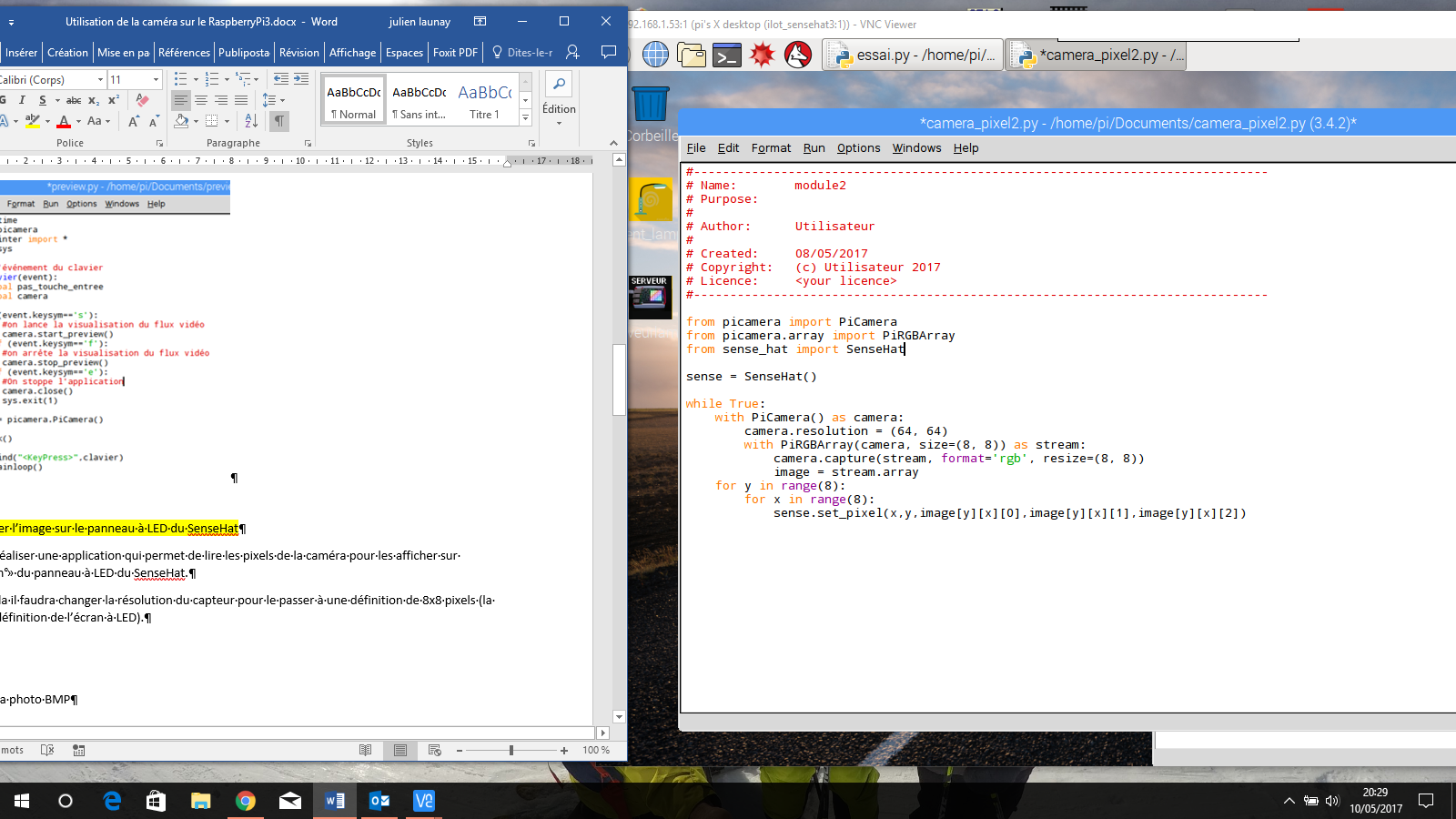
Visualiser l’image sur le panneau à LED du SenseHat

Tu vas réaliser une application qui permet de lire les pixels de la caméra pour les afficher sur « l’écran » du panneau à LED du SenseHat.

Pour cela il faudra changer la résolution du capteur pour le passer à une définition de 8x8 pixels (la même définition de l’écran à LED).

A toi de compléter le programme avec une méthode du SenseHat pour allumer un pixel codé en RGB.

On utilisera la liste image[y][y][0] pour obtenir la couleur rouge d’un pixel de coordonnées (x,y), et image[y][y][1] pour une autre couleur de RGB.



Complète ici la commande manquante

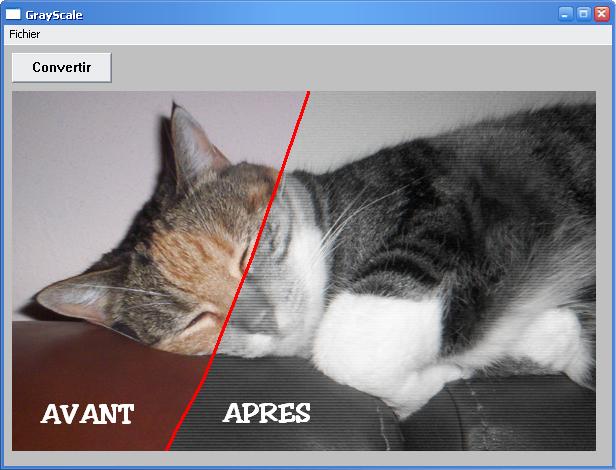
Convertir une image couleur en niveau de gris

Tu utilises des caméras couleurs. Tu as vu comment lire la couleur d’un pixel de ta caméra.

Les trois couleurs (rouge, vert, bleu) sont alors utilisées en couleurs additives. Ce codage est utilisé dans les écrans.

Fais des essais avec ce simulateur pour voir comment obtenir différentes teintes de gris à partir des trois couleurs primaires R,V,B.

Quelle relation doit-on avoir en la niveau de R,V et B ?

[](http://www.profil-couleur.com/lc/006-synthese-additive/melange-synthese-additive.html)

En fait on peut utiliser cette relation classique en traitement d’image.

Niveau\_gris = 0.299 Rouge + 0.587 Vert + 0.114 Bleu au lieu d’une simple moyenne entre R,V,B (Niveau\_gris = (R+V+B)/3) pour prendre en compte la sensibilité de l’œil humain.

Réalise alors un algorithme et ton programme pour modifier le programme précédent afin d’afficher en niveau de gris l’image sur les LED du SenseHat.

Affichage sur l’écran du RaspberryPi d’une photo

Pour cela tu vas utiliser une nouvelle bibliothèque tkinter qui te permettra d’afficher un pixel dans une fenêtre créé sur l’écran.

Tu reprendras le programme vu précédemment pour compléter le programme proposé ci-après.

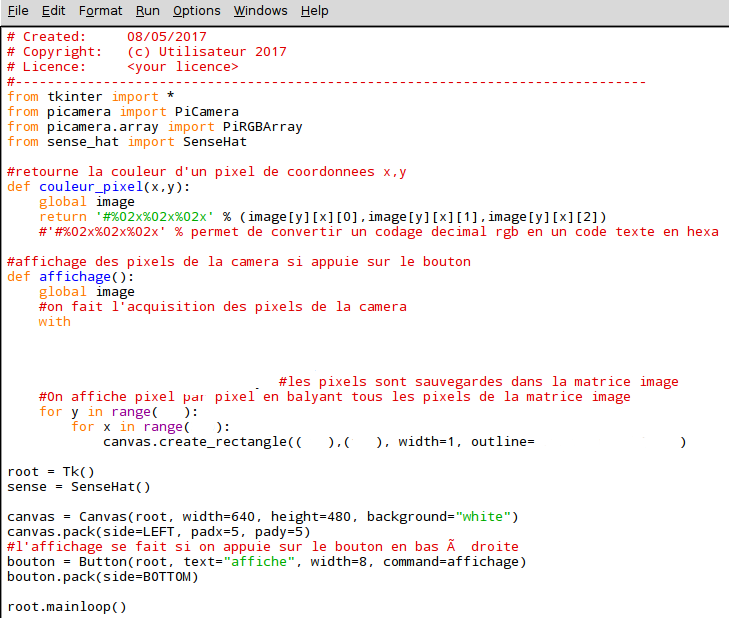
Pour afficher un pixel, tu utiliseras une méthode de dessiner une rectangle d’un seul pixel de côté avec la commande :

Canvas.create.rectangle((coordonnée\_x\_point\_départ,coordonnée\_y\_point\_départ),

(coordonnée\_x\_point\_arrivée,coordonnée\_y\_point\_arrivée), width=1, outiline=couleur\_du\_pixel\_x\_y)

Tu n’oublieras pas de modifier la résolution pour passer en 64x64 pixels pour ta caméra.

Tu supprimeras les commandes size et resize.



A compléter

Recommence ton travail, mais cette fois-ci avec un affichage en noir et blanc.

Détection d’un mouvement.

Le principe est simple. On travaillera en niveaux de gris. Il faudra donc transformer l’image rgb en niveau de gris.

On prend deux images à deux instants t1 et t2

[](http://www.florianpaquet.com/media/uploads/blog/detection-mouvements/frame1_team-member.png)

On fait la soustraction de la couleur de chaque pixel sur les deux images qui ont les même coordonnées x,y dans l’image.

Niveau\_gris\_pixel\_image\_mouvement = Abs(Niveau\_gris\_pixel\_image\_t1 - Niveau\_gris\_pixel\_image\_t2)



On réalise un seuillage de l’image de mouvement.

Si le pixel de comparaison > seuil alors on le transforme en pixel noir (on a détecté un mouvement), sinon il sera codé en pixel blanc (pas de mouvement)

Il suffit de compter le nombre de pixels noirs (ceux qui correspondent au mouvement). En fonction d’un seuil, on considérera qu’il y a eu mouvement.

Premiers essais de streaming avec la caméra sur son réseau

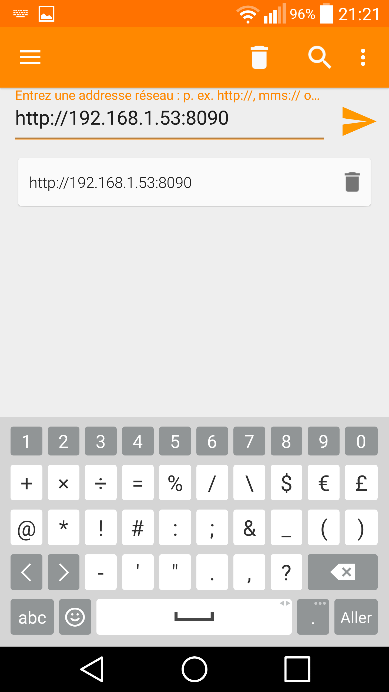
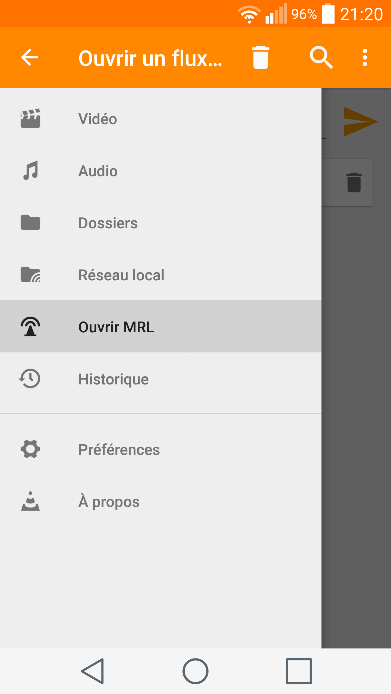
Vérifier que VLC est bien présent

On va rediriger le flux vidéo capturé par la commande raspivid vers le logiciel VLC en exécutant la commande suivante :

raspivid -t 0 -n --width 320 --height 240 -o - | cvlc stream:///dev/stdin --sout '#standard{access=http,mux=ts,dst=:8090}' :demux=h264

Il suffit de lancer VLC sur un client, puis d’ouvrir un flux vidéo sur le réseau via l’adresse IP du raspberrypi suivi du port déclaré lors du lancement de l’appli raspivid. Dans cet exemple, le port est 8090.

On peut utiliser L’application VLC sur son smartphone et visualiser avec 3s de temps de latence la vidéo.

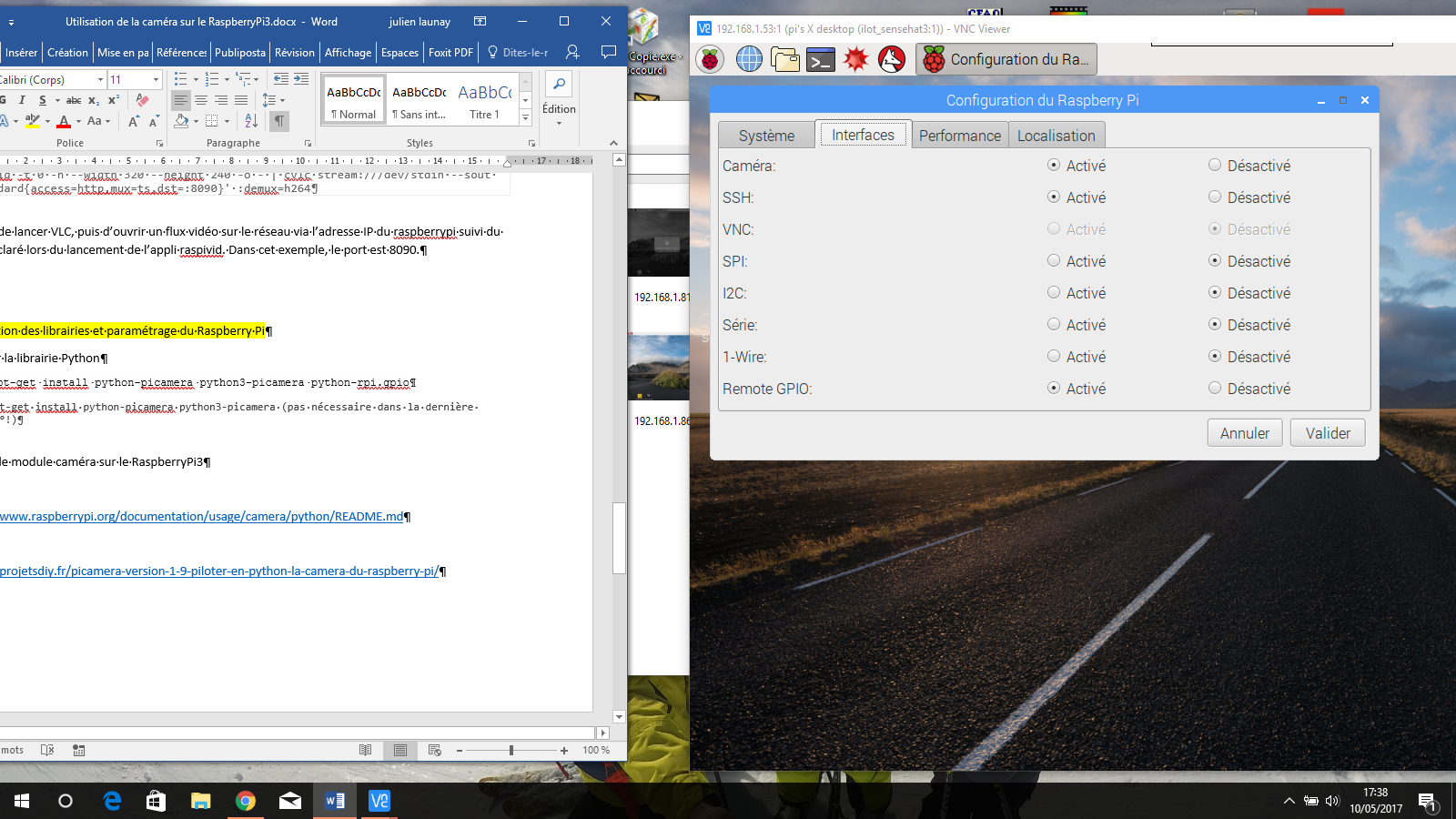


Installation et paramétrage du Raspberry Pi

Installer la librairie Python (plus nécessaire dans les dernières versions de Jessie)

sudo apt-get install python-picamera python3-picamera

Activer le module caméra sur le RaspberryPi3



<https://www.raspberrypi.org/documentation/usage/camera/python/README.md>

<https://projetsdiy.fr/picamera-version-1-9-piloter-en-python-la-camera-du-raspberry-pi/>

<http://www.magdiblog.fr/divers/raspberry-pi-camera-5-facons-de-faire-du-streaming/>

<https://www.element14.com/community/community/raspberry-pi/raspberry-pi-accessories/blog/2015/06/25/getting-to-know-the-raspberry-pi-camera-and-pi-noir>

<https://picamera.readthedocs.io/en/release-0.4/api.html>

<http://www.framboise314.fr/picamera-pour-piloter-integralement-la-camera-de-votre-raspberry-pi-en-python/>