

**Documentation Projet D.R.O.N.E**  
*Équipe de Projet D.R.O.N.E. 2017/2018*

# Contents

<b>1</b>	<b>L'environnement</b>	<b>2</b>
1.1	Lancement Dual Boot Ubuntu / Windows . . . . .	2
1.1.1	Windows . . . . .	2
1.1.2	Ubuntu . . . . .	2
1.1.3	Modification de l'ordre de démarrage . . . . .	2
1.1.4	Configuration de l'ordinateur . . . . .	2
1.2	Contrôle du Rover Spy-C . . . . .	2
1.2.1	Installation de l'API de communication . . . . .	3
1.2.2	Configuration du contrôle par joystick / manette . . . . .	4
1.2.3	Connexion du Rover Spy-C à l'ordinateur . . . . .	4
1.2.4	Lancement du programme RoverPylot . . . . .	4
1.2.5	En savoir plus . . . . .	4
1.3	Contrôle du Drone Parrot Bebop 2 . . . . .	4
<b>2</b>	<b>Lancement des programmes</b>	<b>5</b>
2.1	Sécurité . . . . .	5
2.2	Rover Spy-C . . . . .	5
2.3	Bebop-Autonomy . . . . .	5
<b>3</b>	<b>Annexe</b>	<b>6</b>
3.1	Problèmes courants . . . . .	6
3.1.1	Problème de connexion avec le Rover Spy-C . . . . .	6
3.2	Installations pré-requises . . . . .	6
3.2.1	Installation <i>pip</i> . . . . .	6
3.2.2	Installation <i>PyGame</i> . . . . .	6
3.2.3	Installation <i>OpenCV</i> . . . . .	7
3.2.4	Installation <i>ffmpeg</i> . . . . .	8

# Chapter 1

## L'environnement

### 1.1 Lancement Dual Boot Ubuntu / Windows

Dans cette partie, nous allons voir comment il est possible de démarer l'ordinateur soit sous Ubuntu (une distribution Linux) soit sous Windows.

#### 1.1.1 Windows

Du coté de Windows, l'ordinateur est installé avec la version Windows 10.

#### 1.1.2 Ubuntu

De l'autre coté, Ubuntu 16.04 (Xenial Xerus) est installé pour des raisons de stabilité et de meilleure compatibilité avec la majorité des logiciels écrits pour Linux.

#### 1.1.3 Modification de l'ordre de démarrage

Image du BIOS

#### 1.1.4 Configuration de l'ordinateur

- Processeur : Intel i5
- Mémoire-vive : 8 Go RAM
- Carte Graphique :
- Stockage : 128 Go SSD + 1 To HD

### 1.2 Contrôle du Rover Spy-C

Le Rover Spy-C de Logicom peut se contrôler initialement depuis son application dédiée (disponible à l'adresse suivante sur le Play Store sous Android. Mais depuis peu, Simon Levy est parvenu faire communiquer un ordinateur avec les Rover Brookstone.



Figure 1.1: Rover Spy-C de Logicom

Lien projet GitHub : <https://github.com/simondlevy/RoverPylot>

Un projet reprenant celui de RoverPylot a été repris pour communiquer exclusivement avec le Rover Spy-C de Logicom, le projet *SpyCPylot*.

Lien projet GitHub : <https://github.com/kolas001/SpyCPylot>

### 1.2.1 Installation de l'API de communication

Concernant l'installation de l'API, il est important d'installer les programmes dépendants avant de lancer l'API. Sans ça, certains paquets seront définis manquants. Nous vous conseillons d'utiliser le paquet *Python2.7* pour faire fonctionner l'API.

#### Installation des programmes dépendants

Les paquets nécessaires à la bonne installation de *SpyCPylont* sont :

- *OpenCV*
- *PyGame*
- *ffmpeg*

Les instructions pour l'installation de chacun de ces paquets seront présentés en annexe.

## **Téléchargement de *SpyCPylot***

Téléchargement du projet : <https://github.com/kolas001/SpyCPylot/archive/master.zip>

Extraire l'archive dans ses dossiers

## **Installation de *SpyCPylot***

1. Se placer dans le dossier *SpyCPylot*

```
cd Téléchargements  
cd SpyCPylot
```

2. Lancer l'installation de l'API Python

```
sudo python setup.py install
```

### **1.2.2 Configuration du contrôle par joystick / manette**

### **1.2.3 Connection du Rover Spy-C à l'ordinateur**

### **1.2.4 En savoir plus**

Pour modifier les différents paramètres des scripts Documentation : How I hacked Brookstone-Rover-20

## **1.3 Contrôle du Drone Parrot Bebop 2**

## Chapter 2

# Lancement des programmes

### 2.1 Sécurité

### 2.2 Rover Spy-C

#### 2.2.1 Lancement du programme RoverPilot

1. Connecter la manette à l'ordinateur
2. Lancer le programme *jstest - gtk*
3. Verifier que la manette soit bien reconnue par le programme
4. Vérifier que la carte des touches soit correcte
5. Lancer la commande suivante dans le terminal pour lancer le script python

```
sudo python ps3rover20.py
```

6. Une fenêtre devrait se lancer avec le flux provenant du rover

### 2.3 Bebop-Autonomy

# Chapter 3

## Annexe

### 3.1 Problèmes courants

#### 3.1.1 Problème de connection avec le Rover Spy-C

Voir la documentation du Rover Spy-C

### 3.2 Installations pré-requises

#### 3.2.1 Installation *pip*

*pip* est un système de management de paquets écrit en *Python*. Il est possible grâce à lui il est possible d'installer et desinstaller des paquets.

Mettre à jour son système d'exploitation et tous ses paquets

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

Installation du paquet *pip*

```
$ sudo python get-pip.py
```

Vérification de l'installation de *pip*

```
$ pip -V
```

#### 3.2.2 Installation *PyGame*

Installation du paquet *PyGame*

```
$ sudo apt-get install python-pygame
```

Lancement d'un jeu exemple avec *PyGame*

```
$ python -m pygame.examples.aliens
```

Vérification de l'importation du paquet *PyGame* depuis *python*

```
$ python import pygame
```

### 3.2.3 Installation *OpenCV*

les instructions suivantes proviennent du site web : *pyimagesearch.com*

Installation des paquets pré-requis

```
$ sudo apt-get install build-essential cmake pkg-config
$ sudo apt-get install libjpeg8-dev libtiff5-dev libjasper-dev libpng12-dev
$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
$ sudo apt-get install libxvidcore-dev libx264-dev
$ sudo apt-get install libgtk-3-dev
$ sudo apt-get install libatlas-base-dev gfortran
$ sudo apt-get install python2.7-dev python3.5-dev
```

Récupération du paquet *OpenCV*

```
$ cd ~
$ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip
$ unzip opencv.zip
$ wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip
$ unzip opencv_contrib.zip
```

Création de l'environnement virtuel Python

```
$ pip install virtualenv virtualenvwrapper
$ sudo rm -rf ~/get-pip.py ~/.cache/pip
$ export WORKON_HOME=$HOME/.virtualenvs
$ source /usr/local/bin/virtualenvwrapper.sh
$ echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.bashrc
$ echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.bashrc
$ echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.bashrc
$ source ~/.bashrc
$ mkvirtualenv cv -p python2
$ pip install numpy
$ workon cv
$ cd ~/opencv-3.1.0/
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE
-D CMAKE_INSTALL_PREFIX=/usr/local
-D INSTALL_PYTHON_EXAMPLES=ON
-D INSTALL_C_EXAMPLES=OFF
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules
-D PYTHON_EXECUTABLE=~/.virtualenvs/cv/bin/python
-D BUILD_EXAMPLES=ON ..
$ make -j4
$ sudo make install
$ sudo ldconfig
```

Configuration de *OpenCV* pour *Python2.7*

```
$ ls -l /usr/local/lib/python2.7/site-packages/
$ -rw-r--r-- 1 root staff 2016608 Sep 15 09:11 cv2.so
```



```

adrian@pyimagesearch: ~/opencv-3.1.0/build
[ 98%] Linking CXX executable ../../bin/tapi-example-hog
[ 98%] Built target example_tapi_hog
Scanning dependencies of target example_tapi_ufacedetect
[ 98%] Building CXX object samples/tapi/CMakeFiles/example_tapi_ufacedetect.dir/ufacedetect.cpp.o
[ 98%] Linking CXX executable ../../bin/tapi-example-ufacedetect
[ 98%] Built target example_tapi_ufacedetect
Scanning dependencies of target example_tapi_clahe
[ 98%] Building CXX object samples/tapi/CMakeFiles/example_tapi_clahe.dir/clahe.cpp.o
[ 98%] Linking CXX executable ../../bin/tapi-example-clahe
[ 98%] Built target example_tapi_clahe
Scanning dependencies of target example_tapi_tvli_optical_flow
[ 98%] Building CXX object samples/tapi/CMakeFiles/example_tapi_tvli_optical_flow.dir/tvli_optical_flow.cpp.o
[ 98%] Linking CXX executable ../../bin/tapi-example-tvli_optical_flow
[ 98%] Built target example_tapi_tvli_optical_flow
Scanning dependencies of target example_tapi_pyrlk_optical_flow
[ 98%] Building CXX object samples/tapi/CMakeFiles/example_tapi_pyrlk_optical_flow.dir/pyrlk_optical_flow.cpp.o
[ 98%] Linking CXX executable ../../bin/tapi-example-pyrlk_optical_flow
[ 98%] Built target example_tapi_pyrlk_optical_flow
Scanning dependencies of target example_tapi_camshift
[100%] Building CXX object samples/tapi/CMakeFiles/example_tapi_camshift.dir/camshift.cpp.o
[100%] Linking CXX executable ../../bin/tapi-example-camshift
[100%] Built target example_tapi_camshift
Scanning dependencies of target example_tapi_squares
[100%] Building CXX object samples/tapi/CMakeFiles/example_tapi_squares.dir/squares.cpp.o
[100%] Linking CXX executable ../../bin/tapi-example-squares
[100%] Built target example_tapi_squares
(cv) adrian@pyimagesearch:~/opencv-3.1.0/build$

```

Figure 3.1: Compilation du paquet *OpenCV*

```

$ cd ~/.virtualenvs/cv/lib/python2.7/site-packages/
$ ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so

```

Vérification de l'installation du paquet *OpenCV*

```

$ cd ~
$ workin cv

```

### 3.2.4 Installation *ffmpeg*

```

$ sudo apt-get install ffmpeg

```

```
adrian@pyimagesearch: ~/opencv-3.1.0/build
-- Use IPP Async: NO
-- Use VA: NO
-- Use Intel VA-API/OpenCL: NO
-- Use Eigen: NO
-- Use Cuda: NO
-- Use OpenCL: YES
-- Use custom HAL: NO
--
-- OpenCL:
-- Version: dynamic
-- Include path: /home/adrian/opencv-3.1.0/3rdparty/include/opencv/1.2
-- Use AMDFFT: NO
-- Use AMDBLAS: NO
--
-- Python 2:
-- Interpreter: /home/adrian/.virtualenvs/cv/bin/python (ver 2.7.12)
-- Libraries: /usr/lib/x86_64-linux-gnu/libpython2.7.so (ver 2.7.12)
-- numpy: /home/adrian/.virtualenvs/cv/local/lib/python2.7/site-pack
ages/numpy/core/include (ver 1.11.1)
-- packages path: lib/python2.7/site-packages
--
-- Python 3:
-- Interpreter: /usr/bin/python3 (ver 3.5.2)
--
-- Python (for build): /home/adrian/.virtualenvs/cv/bin/python
--
-- Java:
-- ant: NO
-- JNI: NO
-- Java wrappers: NO
```

Figure 3.2: Compilation du paquet *OpenCV*

```
adrian@pyimagesearch: ~
adrian@pyimagesearch:~$ workon cv
(cv) adrian@pyimagesearch:~$ python
Python 2.7.12 (default, Jul 1 2016, 15:12:24)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'3.1.0'
>>>
```

Figure 3.3: Vérification de l'installation du paquet *OpenCV*