

Documentation Raspberry

Baptiste BEHELLE

Septembre 2021

Table des matières

1	Notes	1
2	Package utilisé	1
3	Setup du Raspberry	2
4	Connexion entre ROS et Raspberry	5
5	Lancer la caméra et Mavros	5
6	Calibrer la caméra	5

1 Notes

Ce document fonctionne aussi bien sur Raspberry 0 ou 3

2 Package utilisé

Pour communiquer avec le contrôleur via le protocole MAVLINK, il faut installer et utiliser le package Mavros ainsi que les extensions pour utiliser l'optitrack par exemple :

<http://wiki.ros.org/mavros>

http://wiki.ros.org/mavros_extras

Pour utiliser la caméra sur le port SPI on utilise le package raspicam_node :

https://github.com/UbiquityRobotics/raspicam_node

Il faut également le module de calibration pour les caméras :

http://wiki.ros.org/camera_calibration

3 Setup du Raspberry

Pour faire fonctionner le drone on met un raspberry sur ce dernier pour le connecter à notre partie de ros et lancer le module mavros sur le rpi part le port série.

On met donc Raspbian buster (lite si pas besoin d'écran) sur une carte sd.

Puis, on installe ROS Melodic ou supérieur depuis les sources sur le Raspberry. On peut suivre le tutoriel suivant pour le faire :

<http://wiki.ros.org/melodic/Installation/Source>

On crée ensuite un espace de travail et on compile nos packages depuis les sources exemple avec Mavros :

Source Installation

This installation assumes you have a catkin workspace located at `~/catkin_ws`. If you don't create one with:

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws
catkin init
wstool init src
```

You will be using the ROS Python tools: *wstool* (for retrieving sources), *roscpp*, and *catkin_tools* (building) for this installation. While they may have been installed during your installation of ROS you can also install them with:

```
sudo apt-get install python-catkin-tools python-roscpp-generator -y
```

TIP

While the package can be built using **catkin_make** the preferred method is using **catkin_tools** as it is a more versatile and "friendly" build tool.

If this is your first time using *wstool* you will need to initialize your source space with:

```
$ wstool init ~/catkin_ws/src
```

Now you are ready to do the build

1. Install MAVLink:

```
# We use the Kinetic reference for all ROS distros as it's not distro-specific and up to
roscpp_generator --roscpp kinetic mavlink | tee /tmp/mavros.roscpp
```

2. Install MAVROS from source using either released or latest version:

- Released/stable

```
rosinstall_generator --upstream mavros | tee -a /tmp/mavros.rosinstall
```

- Latest source

```
rosinstall_generator --upstream-development mavros | tee -a /tmp/mavros.rosinstall
```

```
# For fetching all the dependencies into your catkin_ws,
# just add '--deps' to the above scripts, E.g.:
#   rosinstall_generator --upstream mavros --deps | tee -a /tmp/mavros.rosinstall
```

3. Create workspace & deps

```
wstool merge -t src /tmp/mavros.rosinstall
wstool update -t src -j4
rosdep install --from-paths src --ignore-src -y
```

4. Install [GeographicLib](#) datasets:

```
./src/mavros/mavros/scripts/install_geographiclib_datasets.sh
```

5. Build source

```
catkin build
```

6. Make sure that you use setup.bash or setup.zsh from workspace.

```
#Needed or rosrn can't find nodes from this workspace.
source devel/setup.bash
```

(cf doc de px4 autopilot : https://docs.px4.io/master/en/ros/mavros_installation.html)

On peut utiliser la même méthode pour nos autres packages en les mettant dans le même workspace ou en en créant un à chaque fois.

4 Connexion entre ROS et Raspberry

Pour connecter ROS avec le Raspberry il faut que le raspberry soit sur le même réseau que le pc supportant Roscore.

Puis, on doit entrer 2 commandes dans le raspberry :

```
export ROS_MASTER_URI="udp://<adresse ip pc>:11311"
pour indiquer l adresse de ROS sur le reseau
```

```
export ROS_IP=<adresse ip rpi>
pour envoyer les donnees avec l adresse complete et pas le hostname
pour regler les problemes de DNS.
```

5 Lancer la caméra et Mavros

Pour lancer la caméra :

```
roslaunch raspicam_node camerav2_1280x720.launch
```

Pour lancer mavros :

```
roslaunch mavros px4.launch fcu_url:="/serial0" name:="drone_1"
```

6 Calibrer la caméra

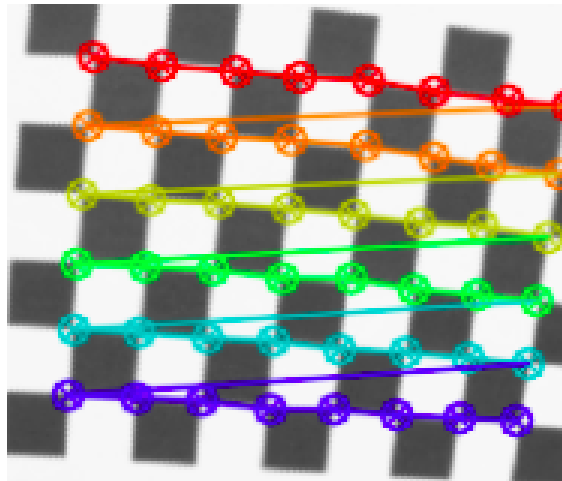
Le but de la calibration est de réduire la distorsion des droites par la caméra en réglant ses paramètres en interne.

On peut lancer la calibration de la façon suivante :

```
roslaunch camera_calibration cameracalibrator.py —size 8x6 —square 0.074
image:=/raspicam_node/image camera:=/raspicam_node
```

Il existe un tutoriel officiel ROS qui explique mieux comment faire :
http://wiki.ros.org/camera_calibration

Mais pour résumer il faut allumer un node et déplacer un échiquier de cases blanches et noires devant la caméra à différentes inclinaisons et distances pour créer un fichier de configuration pour la caméra.



On le met ensuite sur le raspberry et on peut l'utiliser comme paramètre dans nos launch files pour l'utiliser.