# 3.1安装

## 3.1.1安装并编译crazyflie\_ros程序包

新开一个终端

$ mkdir -p ~/crazyflie\_ws/src #自定义工作空间命名

$ cd ~/crazyflie\_ws/src

$ catkin\_init\_workspace

$ git clone https://github.com/whoenig/crazyflie\_ros.git

$ cd crazyflie\_ros

$ git submodule init

$ git submodule update

$ cd ~/crazyflie\_ws

$ catkin\_make

$ source ~/crazyflie\_ws/devel/setup.bash

## 3.1.2扫描并检测crazyflie无人机

打开一个无人机，终端中输入

$ rosrun crazyflie\_tools scan

或

$ rosrun crazyflie\_tools scan --address #后面加上地址

将会启动对范围内的无人机的扫描，并在终端中返回该无人机的uri号，如

radio://0/80/2M/0xE7E7E7E701

# 3.2功能包各部分概要

* **Crazyflie\_Cpp**

包含Crazyradio和Crazyflie涉及到的C++库，在crazyflie\_ros和crazyflie—\_tools中使用，也可以独立于ROS之外使用。

* **Crazyflie\_driver**

包含crazyflie\_server（与所有crazyflies通信）crazyflie\_add（动态添加crazyflies）。

* **Crazyflie\_tools**

包含一些工具，最常用的就是rosrun crazyflie\_tools scan。

* **Crazyflie\_description**

包含一个Crazyflie1.0的一个3D模型，可在rviz中可视化。

* **Crazyflie\_controller**

包含一个简单的PID控制器用于悬停和轨迹导航，可以搭配运动捕捉系统使用，如VICON和Optitrack

* **Crazyflie\_demo**

包含一系列Crazyflie开发实例

·通过遥控手柄建立连接

$ roslaunch crazyflie\_demo teleop\_xbox360.launch uri:=radio://0/100/2M/0xE7E7E7E701

·使用动捕设备（如VICON）使crayflie悬停在（0，0，1）处

其中uri指定Crazyflie的uri，frame指定Crazyflie的tf-frame。启动文件自动运行vicon\_bridge。

$ roslaunch crazyflie\_demo hover\_vicon.launch uri:=radio://0/100/2M frame:=/vicon/crazyflie/crazyflie x:=0 y:=0 z:=1

·对于多个Crazyflies，请确保所有Crazyflies都有不同的地址。

为了获得最好的性能，共享蓝牙的Crazyflies应该使用相同的频道和数据。由于带宽限制，性能会随着每个蓝牙连接的Crazyflies数量而下降，但测试表明可以成功地为每个Crazyradio连接三个Crazyflie。

$ roslaunch crazyflie\_demo multi\_teleop\_xbox360.launch uri1:=radio://0/100/2M/E7E7E7E701 uri2:=radio://0/100/2M/E7E7E7E702 uri3:=radio://0/100/2M/E7E7E7E703