

Crazyflie Manual

Jianxin Sun

3/9/2014

1 Fast setup

1.1 Setup Virtual Machine

- Go to
<https://www.virtualbox.org/>
- Click Downloads on the right
- Download “VirtualBox platform packages” and “VirtualBox 4.3.14 Oracle VM VirtualBox Extension Pack”
- Install the “VirtualBox platform packages (version no.)”
- Install the “VirtualBox (version no.) Oracle VM VirtualBox Extension Pack”
- Go to
<http://wiki.bitcraze.se/projects:virtualmachine:index>
- Download the newest Bitcraze VM file
- Open VirtualBox
- Import the downloaded Bitcraze VM file into VirtualBox (In mac, one can drag and drop the file into the VirtualBox window)
- Now you should see a bar showing the process of importing the Bitcraze VM file
- Once finished, a Bitcraze virtual machine should appear in the right side of VirtualBox software

- Launch the virtual machine.

User name: bitcraze

Password: crazyflie

1.2 Connect to Crazyflie Using PC client

- Once the virtual machine is launched, you should see a Xubuntu desktop fig. 1. If blackscreen, use left win + F to enter fullscreen mode. In full screen mode, if desktop appears, use left win + F to exit fullscreen; if not, relaunch the virtual machine
- There is no need to upgrade anything, double click the “Crazyflie PC client latest” icon on desktop. The client gui should launch as in fig. 2
- Now plug in the radio dongle provided by Crazyflie into your PC’s usb port
- Enable the radio device in virtual machine by selecting it as shown in fig. 3
- Connect your PS3/PS4/Xbox360 controller to mac, enable it in virtual machine just as you enable radio. Make sure that thrust input is 0
- Turn on your mini quad, check section Flight for detail
- Now in PC client GUI, click the Connect button, it should start scan for mini quad
- If your mini quad is on with red led blinking at 2 Hz, you should see it in the list
- If your mini quad is on with red led blinking at 2 Hz and it’s not in the list, click scan button again
- If your mini quad’s red led is not blinking at 2 Hz, go to section Troubleshoot
- Once you see the quad appear in the list as fig. 4, make sure you hold the quad firm in hand and the propellers have clear space to spin, click the connect button
- Now if everything is normal, you should see the data obtained fig. 5

- Now to fly it, go to section Flight.
- To disconnect, click the disconnect button on top

2 Ubuntu setup

To install the PC client on Ubuntu, follow the following link:

http://wiki.bitcraze.se/projects:crazyflie:pc_utils:install

3 Flight

3.1 Before Flight

- read the User Guide:

<http://wiki.bitcraze.se/projects:crazyflie:userguide:index>

This covers the basic concepts such as the meaning of led lights, how to turn on the mini quad, how to find out the front direction of the mini quad, etc.

- check your input device:

http://wiki.bitcraze.se/projects:crazyflie:pc_utils:inputdevices

- Make sure you take necessary safety measures:

<http://wiki.bitcraze.se/projects:crazyflie:safety>

3.2 Pre-Flight test

- You should have connected your mini quad to your PC client right now and getting the live feedback
- You should also have connected your controller to your PC and it is recognized by the PC client
- Now you should fix the mini quad to ground while ensure there are clear space for propellers to rotate.
- Test each stick of your controller to ensure that the control mapping is alright.
- If everything is alright, go to next step

3.3 Fly it

Controlling the mini quad can be difficult at first as it is very agile. One need to practice it a lot to master it. Practice it inside the web area of the lab. Increase the thrust very gently and try not to fly too high above ground as crashes are inevitable in the first few flights. When landing, try to decrease the height slowly. When changing pitch/roll angles, remember to adjust thrust accordingly as part of the thrust will be used to move the quad forward/sideways.

4 Troubleshoot

4.1 regular problems

Check these two pages:

<http://wiki.bitcraze.se/projects:crazyflie:faq>

<http://wiki.bitcraze.se/projects:crazyflie:userguide:troubleshooting>

5 API usage

Some basic info are provided in:

http://wiki.bitcraze.se/projects:crazyflie:pc_utils:pylib

Your controller should be a multi thread program that based on callback functions. The basiclog.py and basicparam.py in example folder demonstrate how to use callback functions to retrieve data from mini quad. The ramp.py shows how to start a thread that takes care of sending control inputs. Make sure you understand these examples before writing your own code.

To run the provided examples that retrieve data, set parameter, control motor, and scan available quad, see the following instruction:

- connect radio to PC and turn on your mini quad
- open up a terminal
- cd to the cfclient project folder, in the virtual machine setup, use this command:
`cd /projects/crazyflie-clients-python/examples/`
- to receive flight data, run:
`python basiclog.py`

- to receive the param list, run:

```
python basicparam.py
```

Remark: the source code of this example has some issues:

To make it work, uncomment “line 174” and comment “line 175”.

Also remember that this program changes the parameter in your quad. So remember to change it back before flying it. A little change in “line 128” should work.

- to control the motors, ensure that quad is fixed to ground and the propellers are free to move, run:

```
python ramp.py
```

- to scan for available quad, run:

```
python scan.py
```

These four example programs demonstrate basic idea of how to get feedback and send out control signal. With these two, one should be able to develop a feedback control loop.

Source code can also be found in:

<https://github.com/bitcraze>

The following links explain some basic ideas about synchronizing between threads, which should be helpful when writing the control loop, as both the control thread and the logging thread will attempt to retrieve/update current flight data. The idea here is to encapsulate the flight data into a class that can only be accessed/modified by one thread at a time, thus preventing the control thread and logging thread from interfering each other.

<http://effbot.org/zone/thread-synchronization.htm>

<https://docs.python.org/3/library/concurrency.html>

Here is a sample code written by Jianxin Sun, for the purpose of building a basic control loop:

<https://github.com/JXS2012/CrazyflieHover>

Remark:

To obtain parameter and log toc, use the PC client gui interface: click view→tabs→log toc(parameter)

6 Improve performance

Due to the limit of data transmit rate, one can only grab limited sensor info from the quad, which is often insufficient for achieving better control over the quad. A possible improvement is to add your feature into the existing quad firmware, allowing access to onboard sensors at a much higher rate. However rewrite the firmware could be a challenging process since only source codes are available. To start, one can go to the following link to study how they write the “stabilizerAltHoldUpdate” function in their firmware:

<https://github.com/bitcraze/crazyflie-firmware/blob/master/modules/src/stabilizer.c>

And there are multiple ways of improving the code, say implementing a Kalman Filter to get better velocity estimate.



Figure 1: Xubuntu Desktop

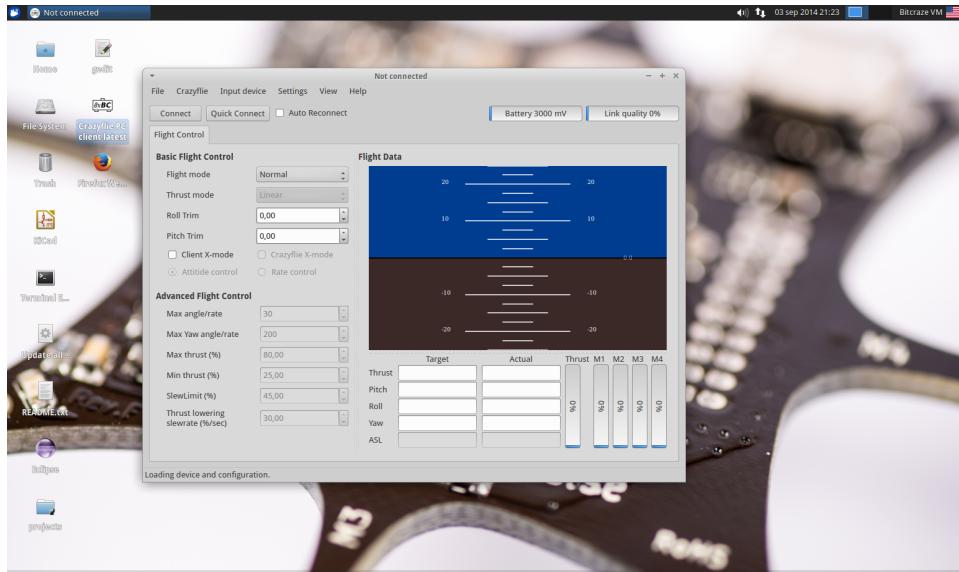


Figure 2: Crazeflie PC client

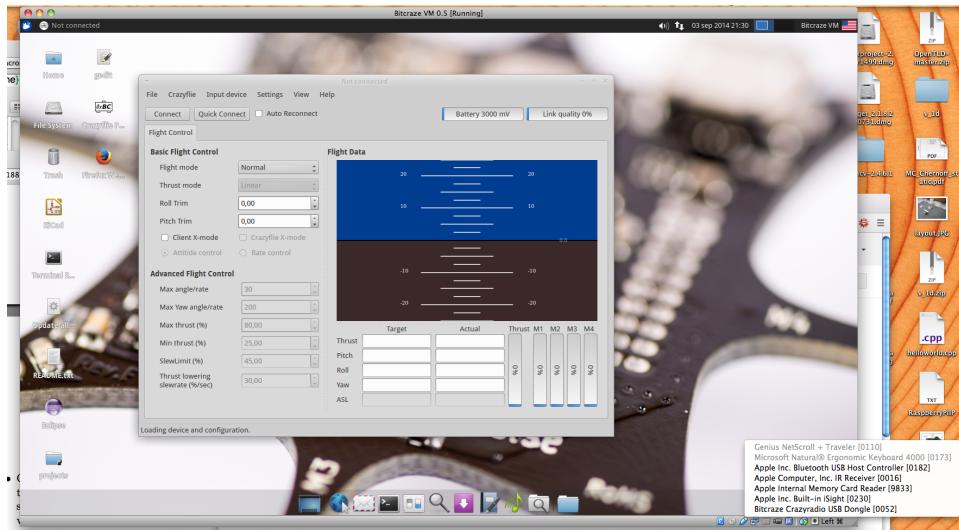


Figure 3: Enable the Bitcraze radio by clicking the usb icon in the bottom of virtual machine window and click the Bitcraze device

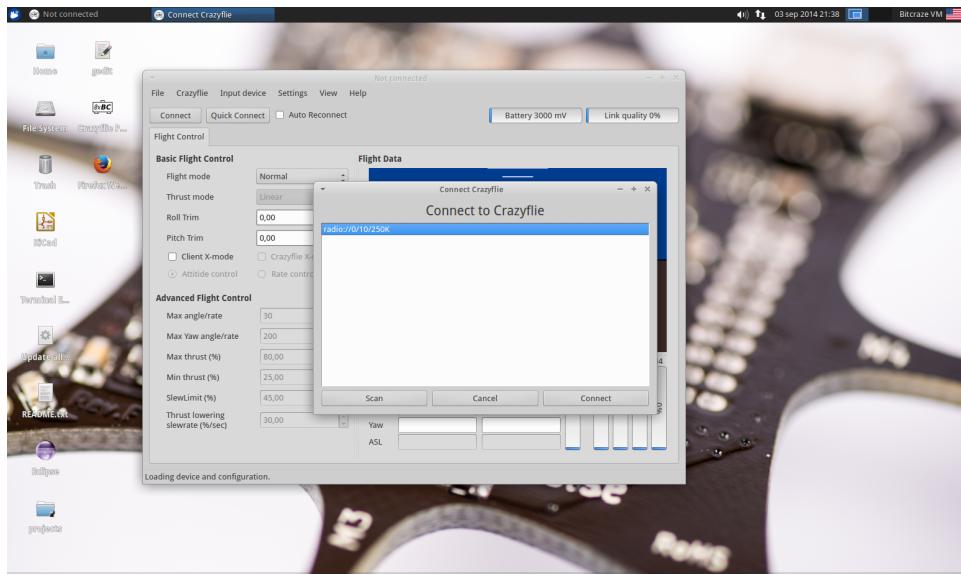


Figure 4: Now the radio://0/10/250K is the quad currently available

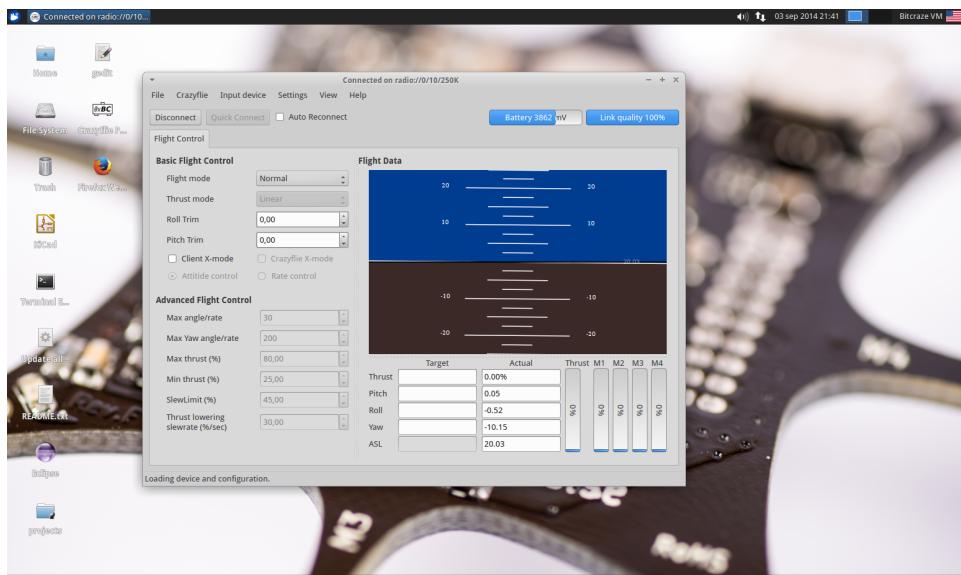


Figure 5: Real time data received from mini quad