

# ME586

Introduction to crazyflie



bitcraze.io

- ◇ Bitcraze is the company which develops and manufactures the Crazyflie, a small, versatile quadcopter for research and education.

### **What makes the crazyflie so special?**

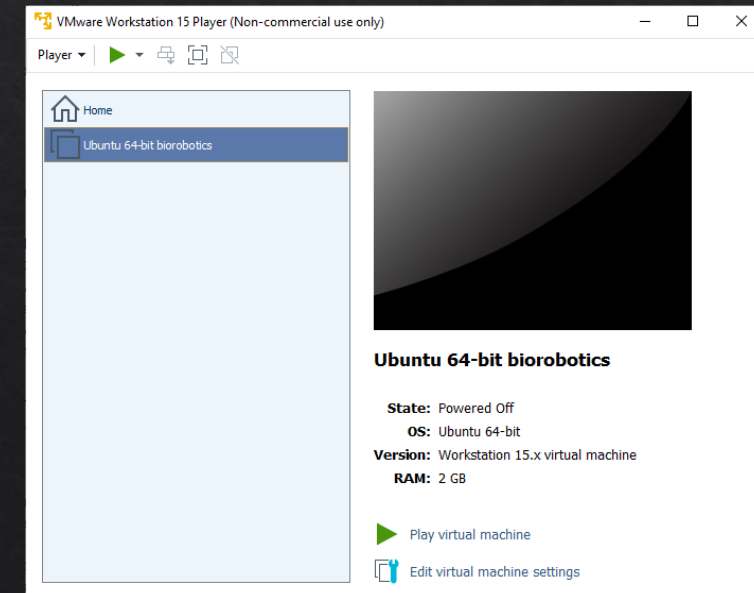
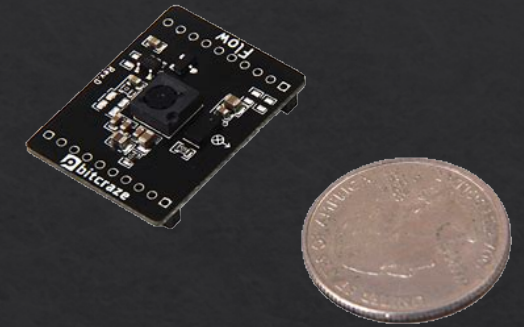
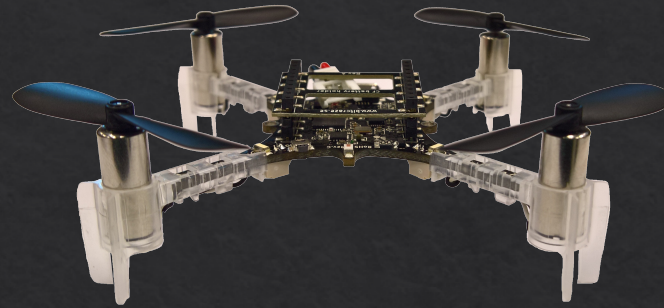
- ◇ Open-source
- ◇ Safe

# Flying the crazyflie

- ◇ Use your phone! (manual flight only)
- ◇ Use a PC and a usb controller (e.g. xbox/playstation controller) (manual flight only)
- ◇ Use a PC and python code (autonomous and/or semi-autonomous flight)

# Your crazyflie

- ❖ Crazyflie 2.1 (most recent version)
- ❖ Optic Flow deck
- ❖ Crazyradio
- ❖ Custom VM
  - ❖ Crazyflie Client (cfclient)
  - ❖ Example python scripts (in ROS)



# The Virtual Machine

- ◇ Different from the bitcraze VM!
- ◇ It is a mirror of the workspace that I run for my research
- ◇ Allows interface with ROS (more about this later)
- ◇ Packages be installed on your own Linux machine (ask me about this if you want to know more!)

# Linux

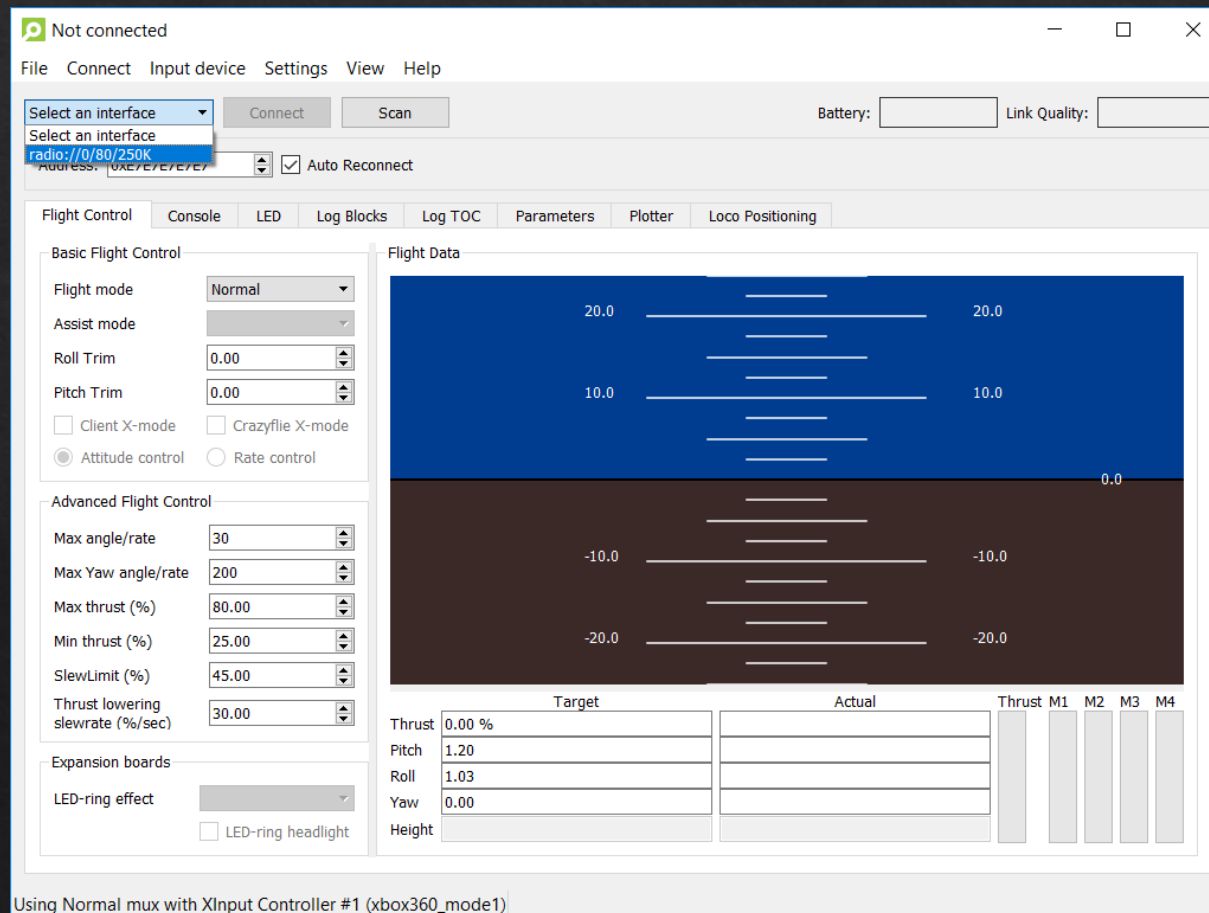
- ◇ The VM is a Linux based system – specifically Ubuntu 18.04

A couple useful commands for navigating within the terminal

- ◇ `cd`
  - ◇ `cd folder_name` – enters the specified folder
  - ◇ `cd folder_name/subfolder_name` – enters subfolder
  - ◇ `cd ..` – goes out one folder level
- ◇ `pwd` – prints out current working directory
- ◇ `ls` – lists all the files in current directory
- ◇ `sudo nano filename.type` – quickly opens the file “filename.type” to be edited within the terminal. E.g. `sudo nano myscript.py`

# Crazyflie client

- ◆ To start the crazyflie client gui, open a terminal and type `cfclient`



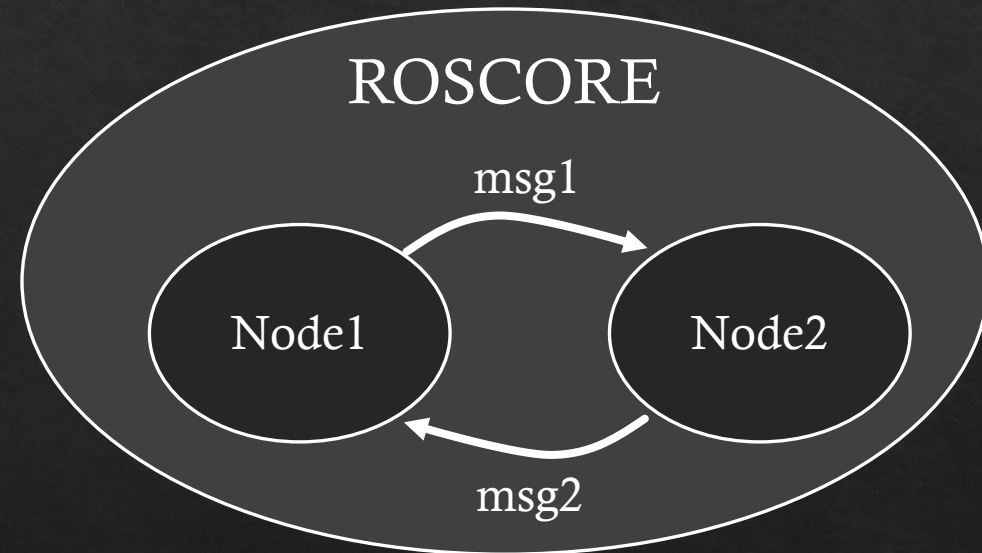
# ROS

◆ The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications.

◆ What I like about ROS:

- ◆ Easy to write and work with “nodes”
- ◆ Easy to communicate between “nodes”
- ◆ Large user database to debug
- ◆ Portable to different robotic platforms

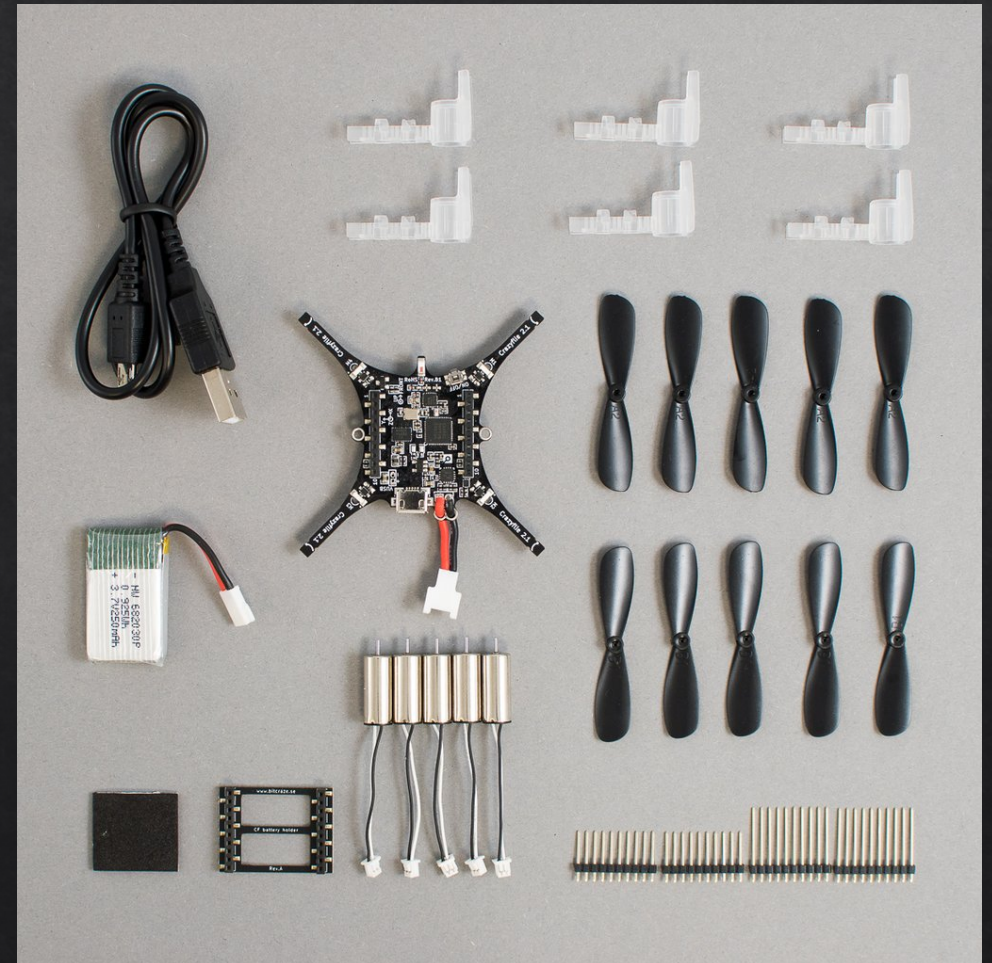
◆ You will NOT need to “learn” ROS, but understanding how and why we use this will be helpful.





# Step 1: Assemble the crazyflie

- ◆ Follow the instructions on [bitcraze.io](http://bitcraze.io)
- ◆ Tutorials > Getting Started with Crazyflie 2.X (STOP at the flying section, do not fly yet!)
- ◆ You are also given a crazyradio, extra battery + charger, Flow Deck, and Prototype Deck
- ◆ Add the Flow deck by using the long header pins and matching up the orientation symbol on the deck to the crazyflie board.



## Step 2: Changing the radio channel on your crazyflie

- ◇ Open cfclient
- ◇ Scan and connect to crazyflie, wait until movement shows up on flight data
- ◇ Go to Connect>Configure 2.X
- ◇ Change the radio channel and write to crazyflie
  
- ◇ Open a terminal and navigate to `catkin_ws/src/rospy_crazyflie/config`
- ◇ Edit the file `config.yaml` so that the crazyflie uri has the correct radio channel

```
crazyflie1 : 'radio://0/81/2M/E7E7E7E7E7'
```



# Step 3: How to fly the crazyflie

- ◆ Plug in crazyradio, connect it to the VM
- ◆ Turn on crazyflie (place on flat surface, press power button, do not move until after beeps)
- ◆ Open two terminals (right click on terminal icon to open second terminal)
- ◆ In one terminal:
  - ◆ `roslaunch rospy_crazyflie default.launch`
  - ◆ Wait until you see that it is “connected” to the crazyflie
- ◆ In the other terminal:
  - ◆ Navigate to the folder `catkin_ws/src/rospy_crazyflie/examples` using the `cd` command
  - ◆ Execute an example program `python takeoff_landing.py`

## Step 4: Examining the code

- ◆ In the file explorer, navigate to one of the python scripts in the examples folder and open it in a text editor (save eag.py for last)

Live example