

# Practical 4: Flying crazyflies

---

[TOC]

**PLEASE READ THIS ENTIRE TUTORIAL IN DETAIL BEFORE YOU START**

## Manual flight

### Before you fly

If you have followed the tutorial, you should have a built crazyflie 2.1 which you can connect to your laptop via a crazyradio or your mobile phone via bluetooth!

We recommend using laptop-based method to fly first, as so far the stablization of crazyflie will only rely on your operation which makes very difficult to control through unprecise mobile phone interface. You could try fly through phones later.

To mannually fly crazyflies through laptops, you need a game controller. Either playstation or xBox could work.

## Crazyradio 2.0 Preparation and Configuration Guide

This guide explains how to prepare and flash the Crazyradio 2.0 firmware, install the necessary USB driver using Zadig (for Windows), set up cfclient in your Miniforge environment, connect and verify your Crazyflie, and change the radio frequency channel.

---

### 1. Prepare & Flash the Crazyradio 2.0 Firmware

**Reference:** [Getting Started with Crazyradio 2.0](#)

#### 1. **Assemble the hardware:**

Screw the provided antenna onto the Crazyradio 2.0 dongle.

#### 2. **Enter bootloader mode:**

Press and hold the onboard button while plugging the dongle into your USB port. You should see a pulsating red LED indicating bootloader mode.

#### 3. **Download and flash firmware:**

- Visit the [Crazyradio2 Firmware Releases \(v1.1\)](#) page.
  - Get the latest compatible firmware file (a `.uf2` file, e.g. named like `crazyradio2-CRPA-emulation-[version].uf2`) from Bitcraze's release page or documentation.
  - Open your file explorer, locate the drive (usually named "Crazyradio2"), and drag & drop the `.uf2` file onto it.
  - Once flashed (the drive disappears), unplug and reinsert the dongle. A brief white LED flash confirms the new firmware is running.
- 

### 2. Install USB Driver Using Zadig for Windows

*This step is not necessary for Linux or macOS.*

### 1. Download and run Zadig:

- Get the [Zadig tool](#) from its official website.
- Open Zadig and enable **List All Devices** from the **Options** menu.

### 2. Install the driver:

- In the device dropdown, select your Crazyradio (often shown as "Crazyradio PA" or similar).
- Choose a libusb driver (e.g. [libusb-win32](#) or [libusbK](#)) and click **Install** (or **Replace Driver**).

*This step is essential for Windows to allow proper USB communication with the Crazyradio 2.0.*

---

## 3. Set Up cfclient in Your Miniforge Environment

**Reference:** [cfclient Installation Instructions](#)

### 1. Install cfclient (or the full crazyflie-clients-python package):

For the stable release where you plan to develop or customize, clone the repository and install in editable mode:

```
git clone https://github.com/bitcraze/crazyflie-clients-python
cd crazyflie-clients-python
pip install -e .
```

### 2. Launch the GUI

Run the following command to to open the cfclient interface.

```
cfclient
```

## 4. Connect and Verify

- Power on your Crazyflie: Make sure your Crazyflie is switched on.
- Scan in cfclient: Click the Scan button in the GUI. The Crazyradio (now emulating a Crazyradio PA) should detect your Crazyflie.
- Troubleshoot if needed: If the Crazyradio isn't listed, recheck the Zadig driver installation and ensure the dongle is correctly flashed and reinserted.

**Note:** Do not fly at this stage before changing the Crazyradio radio frequency channel.

## 5. Changing the Radio Frequency Channel

**Reference:** cfclient Userguide

### 1. Prepare Your Setup:

- Ensure that the Crazyradio 2.0 dongle is correctly flashed and that the proper Zadig driver is installed on your Windows 11 machine (as described above).
- Launch cfclient from your Miniforge terminal.

## 2. Initial Connection:

- Connect to your Crazyflie using the default connection URI. For example:

```
radio://0/80/250k
```

## 3. Open the Configuration Dialog:

- In cfclient, click on the **Connect** menu and select **Configure 2.x** (or **Configure 2.0** if that's what your firmware uses). This opens the configuration dialog for radio-related parameters.

## 4. Set the New Channel:

- Locate the **Radio channel** parameter in the dialog.
- Enter the desired channel value (the Crazyflie supports channels from 0 up to 125, where the channel number sets the frequency offset from 2400 MHz; for instance, channel 15 corresponds to 2415 MHz).
- Note that local regulations might restrict the available frequency ranges, so choose a channel that is legal in your area.
- For best results when using a 2 Mbit/s data rate, channels should be spaced by at least 2 MHz.

## 5. Write and Save the Settings:

- Click the **Write** button to save the new channel value permanently to the Crazyflie's EEPROM.

## 6. Restart and Reconnect:

- Disconnect the Crazyflie, then power-cycle it so that it boots using the new radio channel setting.
- Update the connection URI in cfclient to reflect the new channel. For example, if you set the channel to 50, use a URI like:

```
radio://0/50/2M
```

– Click Scan again and connect; the Crazyflie should now be visible on the new channel.

## Channel Selection Note:

The Crazyflie's channel system works by adding the channel number to a base frequency of 2400 MHz. For example:

- **Channel 20** corresponds to  $2400\text{ MHz} + 80\text{ MHz} = \mathbf{2420\text{ MHz}}$ .
- **Channel 70** corresponds to  $2400\text{ MHz} + 90\text{ MHz} = \mathbf{2470\text{ MHz}}$ .

## Important:

- When operating in the UK, the 2.4 GHz band is legally restricted to approximately 2400 MHz–2483.5 MHz. Therefore, channels should be chosen so that the resulting frequency does not exceed this range.
- Always maintain at least 2 MHz spacing between channels to minimize interference, and coordinate with your peers to ensure no two drones are set to the same channel.

## WARNING

- At this point, you **must** ensure that the crazyflie has external sensing otherwise the drone will be in **manual** flight mode and will have no ability to help you fly or stabilise **it will drift**.
- The joysticks will be mapped directly to roll/pitch/yaw and thrust.
- The joysticks control might be quite sensitive so use **small movements** until you get the hang of it
- You must be as smooth as you can with your joystick control - try not to twitch in the opposite direction when something happens - this results in a hard to recover drone!
- Familiarise yourself with the location of the **emergency stop** if you feel uncomfortable press it at any time (the crazyflies can handle heavy landings)

To understand the coordinate system and roll pitch and yaw see the following link:

- <https://www.bitcraze.io/documentation/system/platform/cf2-coordinate-system/>

When you are ready to fly, **notify an instructor** and place the crazyflie in the middle of your flying space.

## Flying

The instructor will first check all is good with your drone and you have a good setup before you fly.

**Note:** A battery only has around 5 minutes of flying time - keep an eye on it and swap it out if necessary.

Just like being introduced to any new drone, you will first try to get a feel for how the drone will react to your control. Here is a list of actions you can go through to check and learn how your drone flies.

1. **Arming only test** - In this test you will simply arm, wait a second, and then disarm the drone - you will *not* move the sticks. When you arm the drone, the motors will start spinning at minimum RPM, but not takeoff or otherwise move. Disarming the drone will then stop the motors. Does the drone arm and disarm again straight after without needing a restart
2. **Arming ESTOP test** - In this test you will familiarise yourself with, and ensure that the emergency stop button works. Arm the drone and then press the estop. The motors should stop and the GUI should show emergency stop has been pressed.
3. **Floor thrust test** - In this test you will start to familiarise yourself with the thrust control of the controller and response from the drone. You will want to gently increase the thrust, and you should start to see the drone looking like it wants to get off the ground. Try and get a feel for its response! Also worth noting if the drone appears to pivot to one side instead of thrusting equally from all four motors - you may need to compensate a little using the yaw and pitch.
4. **Gentle Takeoff and Landing** - In this test you will attempt to have the drone takeoff and gently land in a controlled fashion. Building upon the previous, you should gently increase thrust until the drone

leaves the ground. When it leaves the ground, try and hold the right thrust level if you can, otherwise reduce the thrust and try and land softly on the ground.

- **Important** A lot of beginner flyers will jerk the stick when it takes off - either plunging it into the ceiling or floor. Try to avoid this by holding your nerve and staying in control.
- **If you feel out of control hit the ESTOP**
- Landing Tip - Thust up a little bit just before you hit the floor, this will reduce the impact on the drone.

5. **Hovering** - Once you are happy and in control with takeoff and landing, you should try and hold the hover for longer, correcting for horizontal drift using the roll/pitch correction.

- **If you feel out of control hit the ESTOP**

6. **Controlled Flight (no yaw)** - Hopefully at this point you should be feeling a lot more in control of the drone (but you probably need a bit of practice!). Now try and slowly fly some shapes (square, circle etc) with the drone always facing away from you (no yaw).

7. **Controlled Flight** - Now try flying the same shapes, but this time using yaw instead of rolling, i.e. always facing in the direction you want to move. It is recommended you try this by first yawing on the spot then moving forward. When you get more confident, you will be able to yaw while moving - this is the closest you'll get to FPV flying in this course!

If working in a pair, don't forget to swap over!

## Assisted manual flight

At this point, you may have noticed that the drone doesn't have any external sensing capability.

You may also have noticed in the GUI and on the control scheme, there is a button labelled *Assist Mode* this will perform an automatic takeoff and activate hover mode as long as its pressed down.

Unlike UGV, Drones need a method of external sensing on top of internal IMU and gyroscope in order to hover in place and not drift, especially if autonomously flying.

What you need to do now is to check which sensor can help you to achieve assisted mannual flight.

For commercial and custom drones there are a variety of sensors available, as will be shown in lectures, and this includes sensors like:

- Cameras (Monocular, Stereo, Depth)
- LIDAR (Single Beam, Wedge and 360)
- Laser Rangefinders
- Optical Flow Cameras
- External Position Tracking (Motion Capture, UWB, etc)

For use with the crazyflies there are several sensor add-on boards available which we may have for the tutorial.

1. [Loco Positioning System](#) (\*\* we are not using this today)

- Hopefully we will have access to the loco positioning system which uses ultra wide band signals from multiple anchors placed around the room.
- This enables a tagging board on the drone to work out the drones position to centimeter level accuracy
- You will have been given a Loco Tag add-on board, this will need to be placed on top of the GPIO stack - it replaces the battery holder!
  - Ensure that it is facing in the correct direction!
- With this the drone will be able to hover and know its location in the real world
- Read more about how it works here:  
<https://www.bitcraze.io/documentation/system/positioning/loco-positioning-system/>
- About General Positioning Systems and crazyflies:  
<https://www.bitcraze.io/documentation/system/positioning/>

For the anchors We will likely need to use the Time Difference of Arrival 3 (TDOA3 Mode) to enable all the drones to fly with lots more anchors - consider why this might be.

## 2. Flow Deck V2

- This is a sensor which can be attached to the bottom of the crazyflie and provides hardware optical flow and height information
- If you don't know what Optical Flow is, it is a method of estimating horizontal motion from a camera image
  - See this opencv tutorial for more information:  
[https://docs.opencv.org/3.4/d4/dee/tutorial\\_optical\\_flow.html](https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html)
- It also has a Time of Flight sensor for measuring its height above ground level
- Together these allow the drone to stably hover in one location, and estimate its location relative to its take-off location!
- Place the sensor on the bottom of the GPIO stack on the underside of the drone
  - Ensure that it is facing in the correct direction!
- Read about the usage here: <https://www.bitcraze.io/documentation/tutorials/getting-started-with-flow-deck/>

## 3. Multi-Ranger Deck

- This board has 5 distance time of flight sensors on it allowing the simple perception of the area around the drone.
- This perception gives the drone the capability to react to its surroundings and perform simple obstacle avoidance, detection, wall following and other tasks.
- It also allows to start working on environment-aware problems like Simultaneous Localization And Mapping (SLAM) algorithms.

## 4. LED Ring Deck (\*\* we are not using these today)

- While not a sensor, its an attachment that can go on the bottom of the deck!
- These will hopefully be used for the light-show challenge
- Note that they go on the bottom and do not work at the same time as the flow deck. As a result, will need the loco positioning system to work properly.

There are a number of other sensors and other details too

- Checkout this [link for a full list of expansion decks](#)

See that link for tips on mounting the decks if confused

## Autonomous Flight

Having hopefully flown the drone manually, you should now be beginning to form some intuitions as to how a drone flies, its features, things to watch out for and so on. Now we move onto starting to consider a key part of this course - which is autonomy.

Autonomy is a bit different for drones over other robots you may have used in the past. Unlike ground vehicles, we have to consider the effects of a 3rd dimension, and need some form of cascaded control to stay in the air. Also whilst the inverse kinematics calculations for arms are not necessary for drones - features such as smooth trajectory generation and following, and mission planning are necessary.

Thankfully, crazyflie offers a simple starting point with a python api for autonomous flight. This will get you back up to speed with programming and testing robots before we delve into more complex autonomous systems which you would see flavours of in the real world. Remember at the end of the day, all of these different frameworks are just tools - we hope that through learning to use these tools, you also absorb and learn more generalisable features of drone autonomy which you could apply to the many other open source and proprietary systems which exist out there!

For crazyflie autonomy, try out the following links which show a simple getting started!

- Flow Deck [tutorial](#)
- Multi-Ranger Deck [tutorial](#)
- Loco Deck [tutorial](#)
- Example Scripts [Link](#)

### **Your goal here is to have the drone fly in some shapes autonomously**

As a bonus, see if you can start to work out how to use the LED Deck!

You will likely need to look up a little of the documentation for the api but I'm sure you can mostly use the examples to work out the best way to program the drones to do what you need. Don't worry about making things too complicated, as we will be quickly moving on to ROS2 for drone control!