

The Bitcraze ecosystem

Goals

An understanding of the Bitcraze ecosystem and how everything fits together

- Hardware
- Firmware
- Clients and libraries
- Use Cases

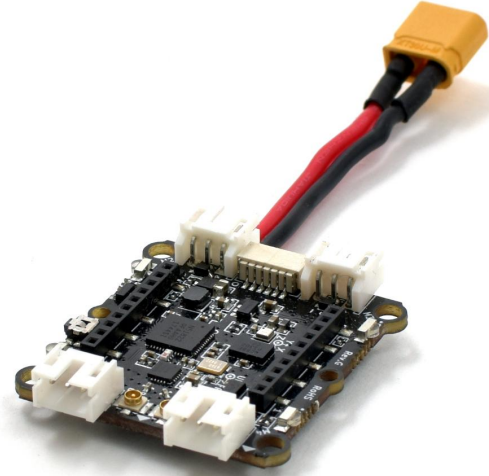
Hardware - Quadcopters - Crazyflie 2.1

- 7 min flight time
- 27 g
- 15 g lifting capacity
- Cortex-M4, 168MHz, 192kb SRAM, 1Mb flash
- 2.4 GHz Radio and BLE
- USB
- Expansion port
- Robust
- Brushed motors



Hardware - Quadcopters - The Bolt

- Specs similar to Crazyflie 2.1
- Custom frame
- Brushless motors
- Higher lifting capacity
- Larger battery



Hardware - Crazyradio PA

- USB dongle
- Long range
- 2.4 GHz
- 250 Kbit/s - 2 Mbit/s
- 125 channels



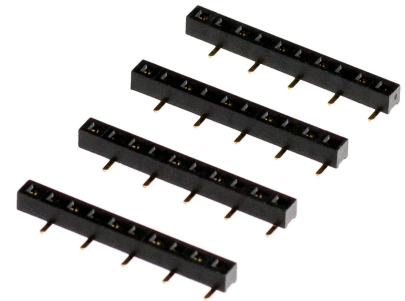
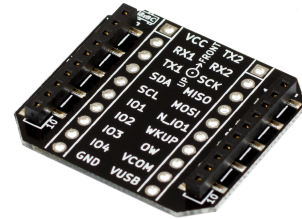
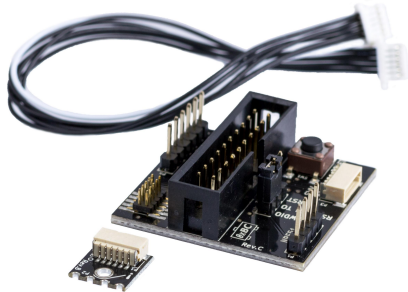
Hardware - Expansion decks

- Sound and light
- Relative positioning
- Absolute positioning
- Development
- Other



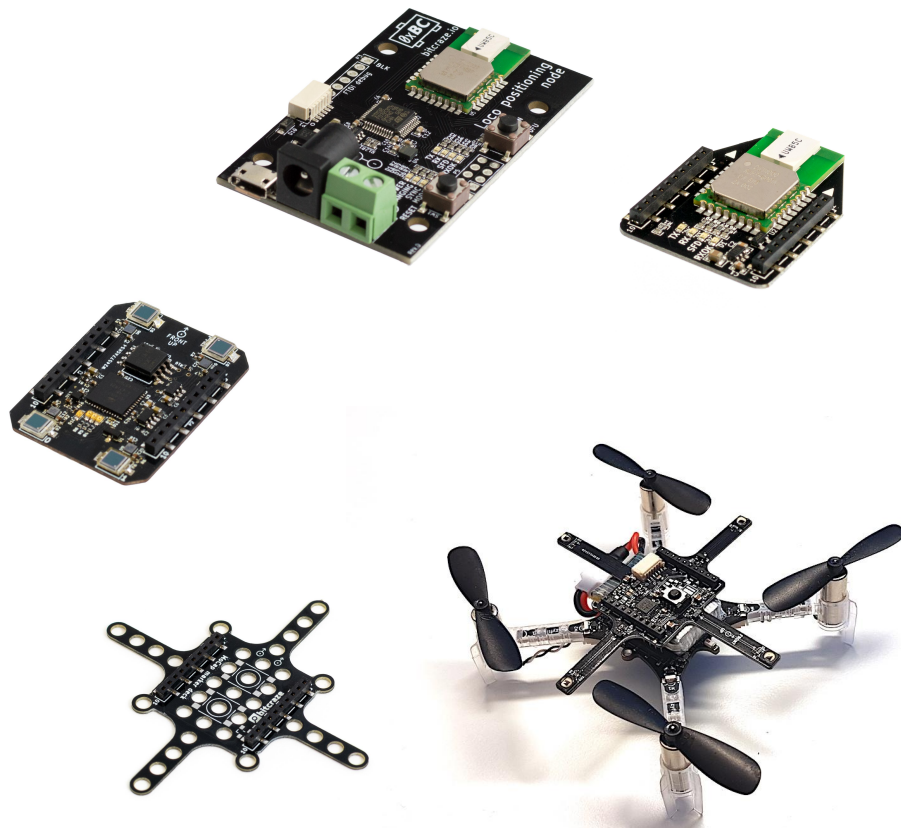
Hardware - development support

- Design your own deck
 - Prototyping deck
 - Breakout deck
- Debug adapter
 - For debugging firmware



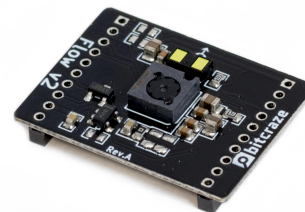
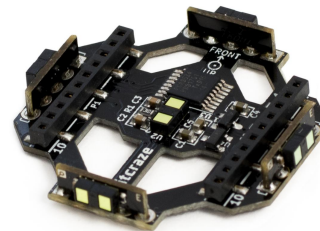
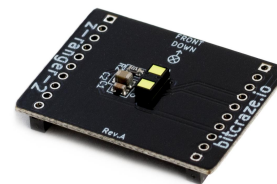
Hardware - absolute positioning

- LPS
 - LPS deck
 - LPS Node (AKA Anchor)
- Lighthouse
 - Lighthouse deck
 - 2 x HTC Vive base station
- Motion capture
 - Marker deck
 - Active marker deck (Qualisys)
 - Motion capture system from any vendor



Hardware - relative positioning

- Z-ranger deck
- Multi-ranger deck
- Optical Flow deck



Hardware - the Roadrunner

- Crazyflie 2.1 + LPS deck - motors
- Generic positioning device for LPS
- Position calculated in the Roadrunner
- Expansion port
- Compatible with firmware, libraries and clients



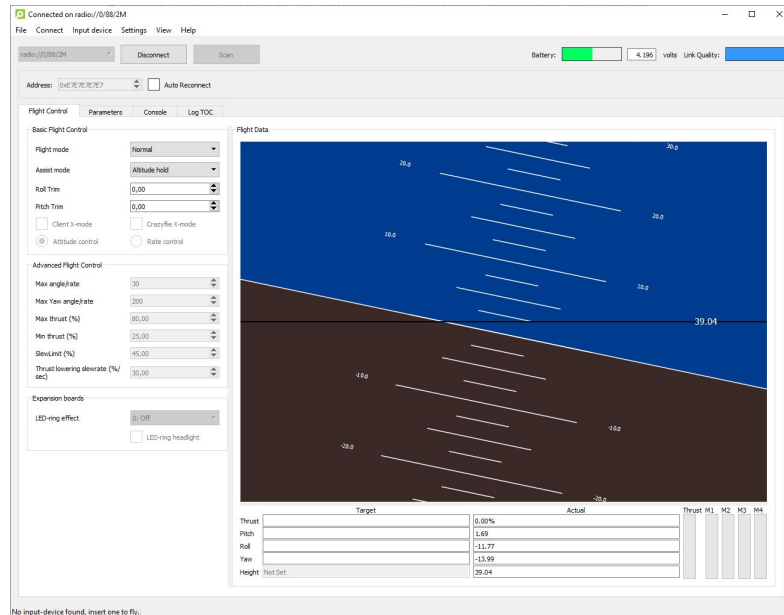
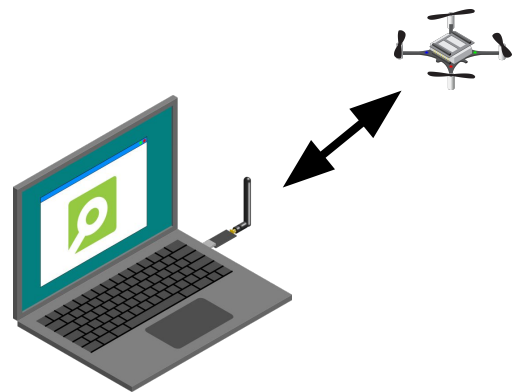
Firmware - Crazyflie

- Open source
- Intended to be modified
- C
- FreeRTOS



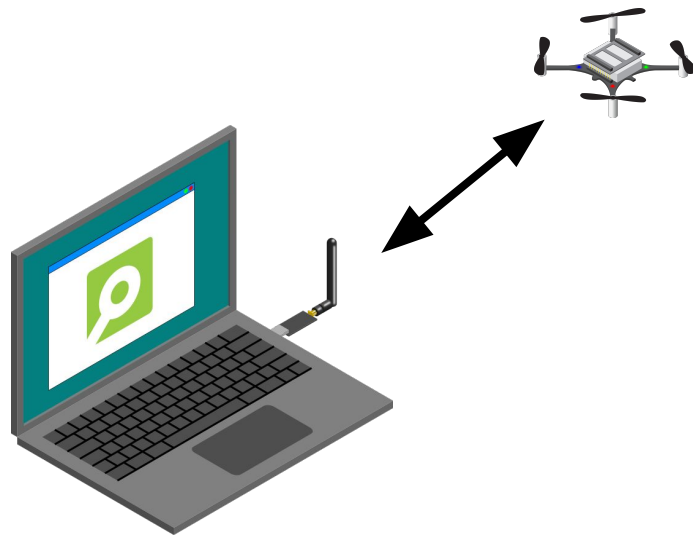
Clients - PC

- Python
- Linux / MacOS / Windows
- Connect to one Crazyflie / Roadrunner
- Manual flight with gamepad
- Examine internal state
- Configure
- Connects via radio or USB



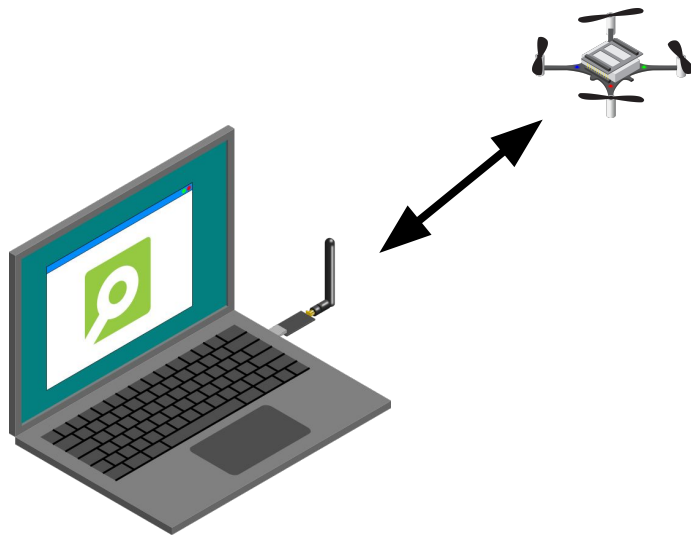
Clients - python library

- Used by client
- Access to full system
- Can be used for scripting
 - Autonomous flight
 - Access to sensor data
 - Configuration and control
 - ...



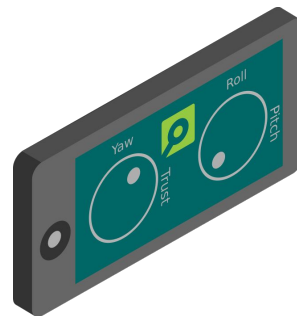
Clients - python library - log / param / mem

- Log
 - expose variables in the Crazyflie.
Gives access to internal state
- Param
 - set variables in the Crazyflie.
Enables real-time configuration and control
- Mem
 - mapping of memory in the Crazyflie to python objects. Useful for complex data structures and larger blobs of data.



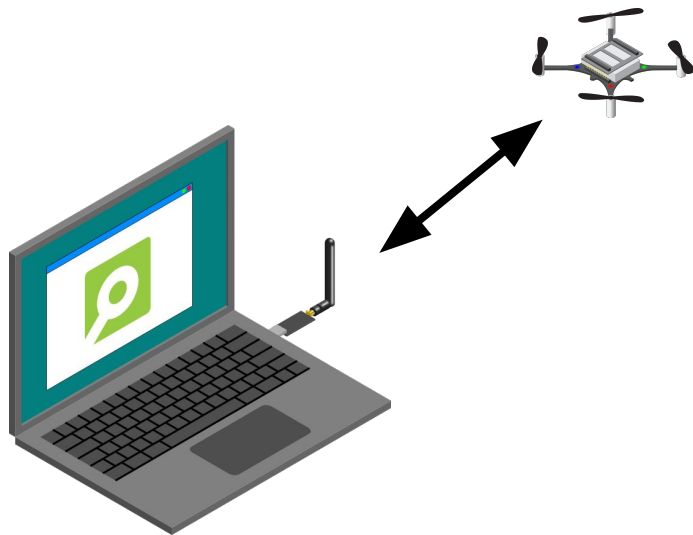
Clients - mobile apps

- Android
- IOS
- Connects via BLE
- Limited functionality
- Manual flight



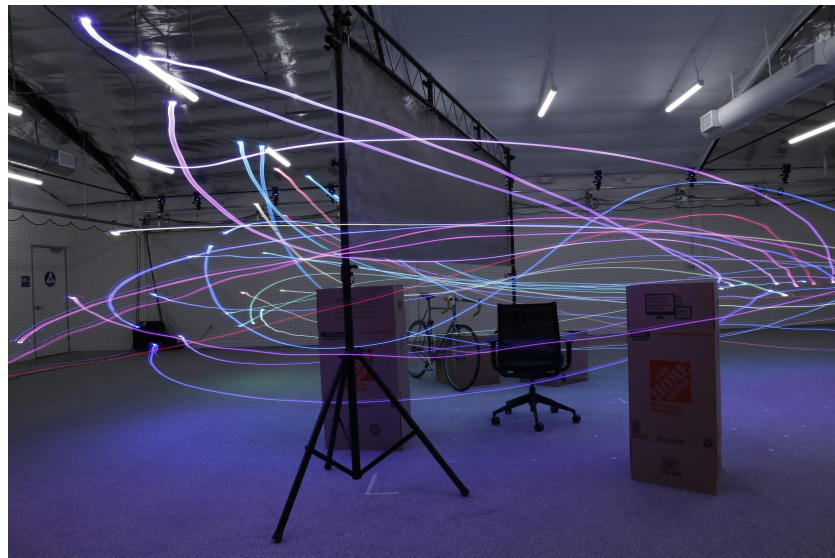
Clients - external libraries

- C, Java, JS...
- Maintained by the community
- Might not be up to date



Clients - CrazySwarm

- Library to simplify swarm support
- Handles large swarms
- Uses ROS
- Originally written for mocap systems but works with other position technologies as well



Use cases - manual flight

- Crazyflie
- PC
 - Crazyradio
 - Gamepad
 - Python client

Use cases - autonomous flight, LPS positioning

- Crazyflie
 - LPS deck
- PC
 - Crazyradio
 - Python script
- 8 x LPS Nodes

Use cases - autonomous flight, mocap positioning

- Crazyflie
 - Marker deck + passive markers
- PC
 - Mocap software
 - Python script
 - Crazyradio

Conclusions

- Everything fits together
- Designed for development
- Open source