CRAZY FLIE

**General Link**:

Forum

https://forum.bitcraze.io/viewtopic.php?f=6&t=2471&p=12460&hilit=Understanding+the+code#p12460

Microprocessor:-

<https://developer.arm.com/products/processors/cortex-m/cortex-m3>

Uses Armv7 CPU  
<https://web.eecs.umich.edu/~prabal/teaching/eecs373-f10/readings/ARMv7-M_ARM.pdf>

**Architecture**

<https://wiki.bitcraze.io/projects:crazyflie2:architecture:index>

It has two microcontrollers and communication between them is controlled using **Syslink** protocol  
<https://wiki.bitcraze.io/doc:crazyflie:syslink:index>

**RF Communication**

There is RF receiver on crazy flie to receive commands from RF sender (Bluetooth, CrazyFlie dongle).

<https://www.bitcraze.io/2012/02/the-crazyradio-dongle/>

**Bluetooth light (BLE) connection**

<https://www.bitcraze.io/2014/08/crazyflie-2-0-bluetooth-low-energy-connectivity/>

nRF51822 supports both BLE and radio connectivity. It uses Nordic Semiconductor soft device BLE stack.

<https://www.nordicsemi.com/News/News-releases/Product-Related-News/Nordic-flagship-Bluetooth-Smart-protocol-stack-enables-sophisticated-Bluetooth-Smart-wearable-hub-networks-with-sensors-that-don-t-need-smartphones>

<https://community.nxp.com/thread/332319>

**CRTP Protocol**

The protocol used to communicate with the Crazyflie is called the Crazy RealTime Protocol, or CRTP in short. It's a simple protocol mainly designed to suit the radio chip used on the Crazyflie but the protocol can also be used on other physical links.

Used from Android client to CrazyFlie communication.

<https://wiki.bitcraze.io/projects:crazyflie:crtp>

**LOGGING**

Logging is complex task in Crazyflie, client can define which variables will get logged in crazyflie during runtime and how frequently will be sent back to client.

<https://wiki.bitcraze.io/projects:crazyflie:firmware:log>

https://wiki.bitcraze.io/doc:crazyflie:crtp:log

**MOTOR Control**The motor is controlled using PWM signals.

<https://oscarliang.com/pwm-ppm-difference-conversion/>

<https://www.arduino.cc/en/Tutorial/PWM>

- One way DC motor is controlled is using PWM, where the signal (square wave) is controlled to be On/Off delivering power to motor when “On” and None (Inductance would kick in) when off.

- That is why we need control mechanism to adjust width of PWM signal to deliver necessary power to control speed of motor.

- So, there would be some delay from calculated acceleration/speed at high level and its reflection into motor.

<https://www.elprocus.com/pulse-width-modulation-pwm/>

**BUS**

<https://en.wikipedia.org/wiki/I%C2%B2C>

I2C bus used to transfer data between microcontroller and peripherals

**CONTROL**

**HOVERING**

Hovering in crazyflie is achieved using Zranger deck, which measures distance from the floor. Sensor used

Sensor API  
<https://www.st.com/content/ccc/resource/technical/document/user_manual/group0/6b/4e/24/90/d8/05/47/a5/DM00279088/files/DM00279088.pdf/jcr:content/translations/en.DM00279088.pdf>

**Facts about sensor:**

Sensor needs to be calibrated by customer because of various temperature condition, other signals, cross talk, etc.

**NRF51822 Documentation**

[**https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.sdk52.v0.9.0%2Fgroup\_\_nrf\_\_gpio.html**](https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.sdk52.v0.9.0%2Fgroup__nrf__gpio.html)

**Sensor is connected using I2C, SPI protocol**  
<https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

<https://learn.sparkfun.com/tutorials/serial-communication>

<https://learn.sparkfun.com/tutorials/i2c>

<https://i2c.wiki.kernel.org/index.php/I2C_Tools> (Less important)

**Future Readings**

[**https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-002-circuits-and-electronics-spring-2007/video-lectures/**](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-002-circuits-and-electronics-spring-2007/video-lectures/)

**Unrelated:** [**https://ocw.mit.edu/courses/sloan-school-of-management/15-084j-nonlinear-programming-spring-2004/lecture-notes/**](https://ocw.mit.edu/courses/sloan-school-of-management/15-084j-nonlinear-programming-spring-2004/lecture-notes/)

[**https://ocw.mit.edu/resources/res-6-010-electronic-feedback-systems-spring-2013/course-videos/**](https://ocw.mit.edu/resources/res-6-010-electronic-feedback-systems-spring-2013/course-videos/)

CrazyFlie Firmware NRF51

Github, Wiki - <https://github.com/bitcraze/crazyflie2-nrf-firmware/blob/master/readme.md>

Travis CI - <https://docs.travis-ci.com/user/getting-started/> for continuous integration, travis runs the build in VM and see if all tests are passing.

Running tests: <https://github.com/bitcraze/crazyflie2-nrf-firmware/blob/master/CONTRIBUTING.md> - ./tools/build/build