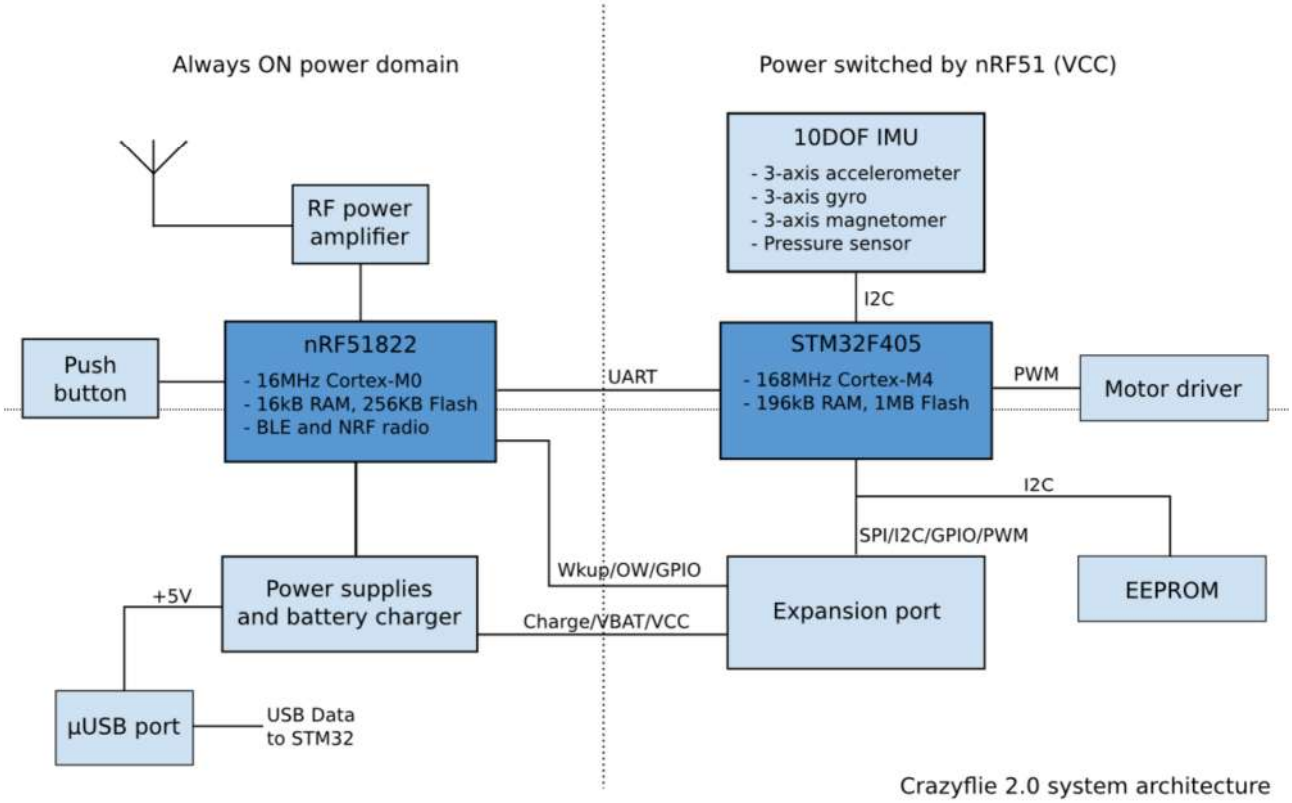
Bitcraze Wiki

Crazyflie 2.0 System Architecture

Crazyflie 2.0 is architectured around 2 microcontrollers:

- A **NRF51**, Cortex-M0, that handles radio communication and power management:
  - ON/OFF logic
  - Enabling power to the rest of the system (STM32, sensors and expansion board)
  - Battery charging management and voltage measurement
  - Master radio bootloader
  - Radio and BLE communication
  - Detect and check installed expansion boards
- An **STM32F405**, Cortex-M4@160MHz, that handles the heavy work of flight control and everything else:
  - Sensor reading and motor control
  - Flight control
  - Telemetry (including the battery voltage)
  - Additional user development

This page aims at explaining a bit how things are working together.



The nRF51822

The two main tasks for the nRF51 is to handle the radio communication and the power management. It acts as a radio bridge (it communicates raw data packet to the STM).

Crazyflie 2.0 use the radio for both CRTP and BLE, but the hardware also supports other protocols like ANT. The CRTP mode is compatible with the Crazyradio USB dongle and it provides a 2Mbit/seconds data link with low latency. Test shows that the latency of the radio link is between 360us and 1.26ms, at 2Mbps without retry and a packet size of respectively 1 and 32 bytes. The minimum achievable latency with Bluetooth is 7.5ms but current implementation is more around 20ms. The main benefit of the CRTP link with the Crazyradio is that it's easily implemented on any system that supports USB host which, makes it the first choice to hack and experiment with the Crazyflie. BLE is implemented mostly with the use case of controlling the Crazyflie 2.0 from a mobile device.

One of the other particularities of the nRF51 chip is that it was designed to run from a coin battery, which means that it is pretty well suited for low energy operation. So the NRF51 is also responsible for power management. It handles the ON/OFF logic which means that the NRF51 is always powered and that different action are possible when pressing the ON/OFF button for a long time (ie. this is used to start the bootloader). It is also possible to wake Crazyflie 2 from one pin of the expantion port, which allows wake-up by an external source.

The STM32F405

The STM32 runs the main firmware. Even though it is started by the NRF51, it acts as a master towards the NRF51. It implements flight control, and all communication algorithm. The expansion port is mainly connected to the STM32 so the driver for expansion boards sits in the STM as well.

The STM32F405 has 196kB of RAM which should be enough for anyone (famous last words…). This is overkill for just the flight controller but it allows for more computationally intensive algorithms, for example sensor fusion between inertial sensors and the GPS data.

Inter-MCU communication

The comminication between the two CPUs is handled by the syslink protocol. It is a simple packet-based protocol we made to have an extensible communication scheme.

Syslink provides messages for carrying all required communication between the CPUs. The STM32 is the master and the NRF51 the slave. As much as possible we try to keep the NRF51 simple and stupid to offload complex algorithm in the STM32.

Example of syslink message are:

- Raw radio packets, to be sent and received
- Power management measurement

projects/crazyflie2/architecture/index.txt · Last modified: 2015-07-15 16:31 (external edit)