Design, Construction & Control of A Micro-UAV Airship

Erez Gotlieb

Supervision Yonatan Mandel

Faculty of Electrical Engineering, Tel Aviv University

Abstract

UAVs, and particularly drones, have become a main topic of research in universities around the world in recent years. They provide a convenient platform for the study of various advanced topics in the field of robotics and engineering. Topics including but not limited to control theory, image processing and of course more recently - computer vision and AI.

Despite the various advantages of micro aerial vehicles (micro UAVs), they suffer a major downside (as opposed to land-based vehicles etc.), flight-time. Most commercial micro UAVs are limited to a flight time of 5 (for small platforms) to 30 (for larger platforms) minutes. This limitation poses challenges in studying, developing and testing new algorithms in a lab-based environment.

The main objective of this project is to provide the TAU-ADL a platform with drastically better flight time performance than any other currently available platform to the lab. This platform shall later serve as a tool for studying and executing advanced algorithms in the fields discussed above.

The project is to be based on the currently open-source available platform to the ADL – the Crazyflie mUAV, together with a commercially available helium balloon.

The project goals are as follows:

1. Demonstrate a dramatic increase in flight time – with potential for scaling.
2. Demonstrate robust control of position and yaw of the platform within a closed space environment.
3. Demonstrate integration of a camera and SLAM position control.

The project has several main stages:

1. Aeronautics Design phase –

   Design and build a prototype of the mUAV, including mechanical and electrical modifications to the platform.

2. Control algorithm design and test –

   Design and test control algorithms that will allow the robust control of position and yaw of the platform in flight.

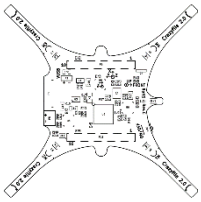
Contents

Abstract 2
Introduction - Crazyflie-2.0 Platform 4
Mechanical specifications 4
Architecture 4
The **nRF51822** 5
The **STM32F405** 6
Inter-MCU communication 6
Aeronautics Design 7
Design considerations 7
Vehicle Dynamics & Motor Mixing 9
Frame Design, Manufacturing & Assembly 10
Electronic Design, Manufacturing & Assembly 12
Control Scheme Modifications 14
Flight Time - Test Results 16
SLAM Integration 16
Overview 16
Calibration 18
Summary 21
Source Files Repositories 21
References 22

Introduction - Crazyflie-2.0 Platform

The Crazyflie is a miniature quadcopter (mUAV) designed and built to be an open source and extensible research platform for universities and “hacker” alike.

Mechanical specifications



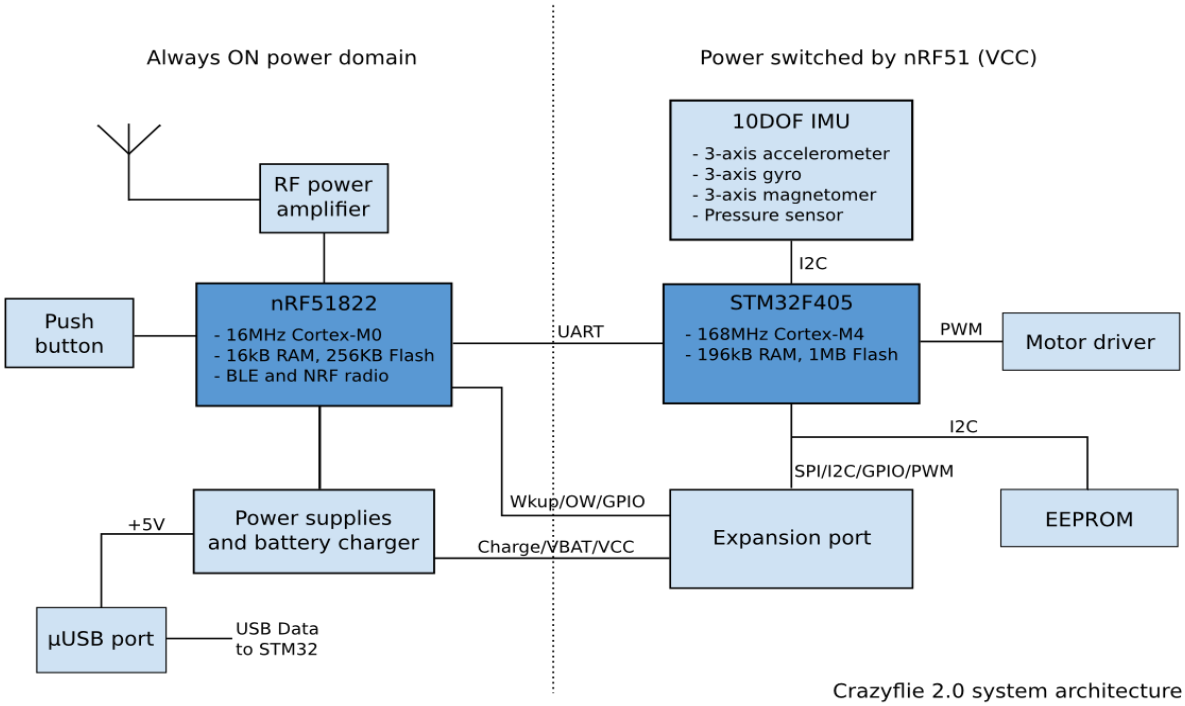
Takeoff weight: **27g**

Size (WxHxD): **92x92x29mm** (motor-to-motor and including motor mount feet)

Architecture

Crazyflie 2.X is based on 2 microcontrollers:

1. **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
2. **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



Crazyflie 2.0 system architecture

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.X 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 (for PC comms) 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.X 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 well suited for low energy operation. 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).

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. 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 communication between the two CPUs is handled by the syslink protocol. It is a simple packet-based protocol 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 algorithms to the STM32.

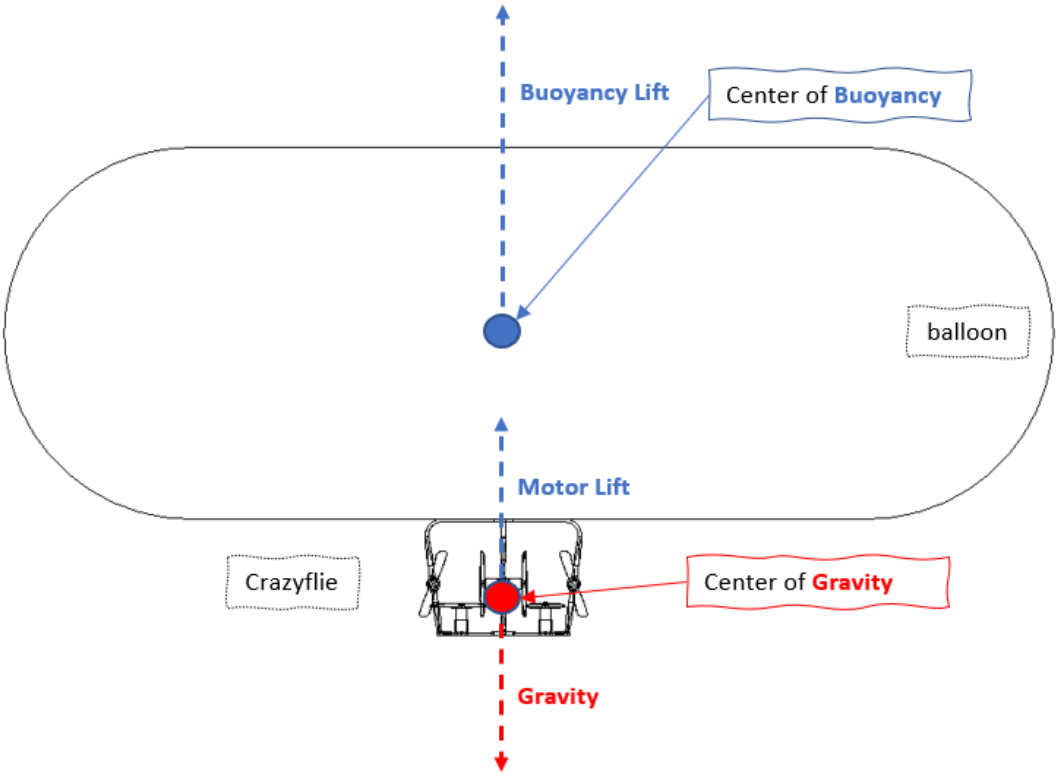
Aeronautics Design

Design considerations

The idea behind the design is to use a helium balloon in order to extend the short flight time of the Crazyflie. With its standard frame and battery, the Crazyflie weighs roughly **30 grams** and can stay airborne for roughly **5-6 minutes**. The balloon will be connected to the newly designed frame and provide an additional lift force in the form of buoyancy.

Assuming linear relations (for the sake of simplicity at first) and assuming most of the Crazyflie's power is spent on the motors, in order to achieve a drastic flight time improvement, we require a proportionally drastic reduction in effective weight (that the motors are lifting). The effective weight is simply the weight of the modified Crazyflie (new configuration) when deducting the "thrust" produced by the balloon.

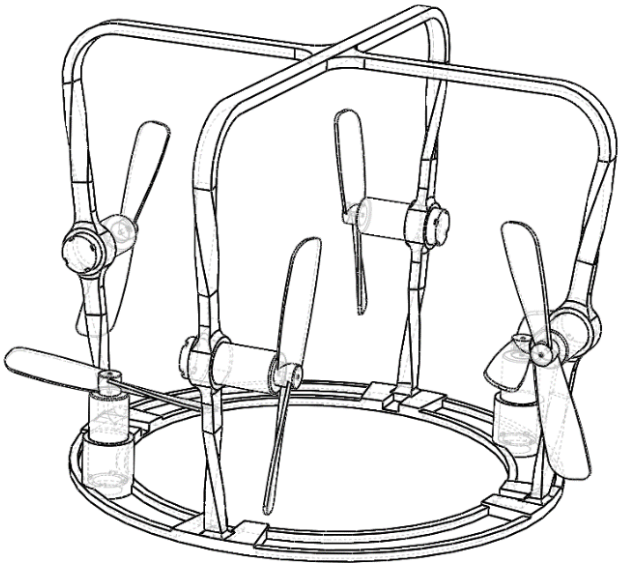
From the above we easily deduce that the number one design goal of the new configuration should be to **minimize weight**.



The new configuration, depicted in the figure above, has entirely new dynamics. In order to achieve robust control of the new vehicle a new motor configuration is required (as opposed to the traditional quadcopter configuration). The configuration constraints were deduced to being as follows:

a. The new dynamics are **naturally stable** (since CG is below CB/CL) and thus translation can only be achieved by application of force in the desired direction.
b. The stable dynamics allow **relinquishing control of the pitch & roll** of the vehicle without compromising stability.
c. The new configuration requires a small amount of lift (the effective weight of the configuration will be downwards i.e. positive). In order to achieve **maximum efficiency** in application of the lift, the lifting motor/s should be placed in an **upright position** (the force they produce should be directly upwards).
d. In order to **maintain symmetry** and cancel the torques produced by the vertical motor/s there should be an even number of them (at least 2) rotating in opposite directions.
e. In order to **maintain symmetry** and cancel torques produced by the lateral motors they should be placed in a symmetric fashion.

The new configuration purposed is depicted in the following drawing and picture:



Vehicle Dynamics & Motor Mixing

The vehicle dynamics in the proposed configurations are governed by the following forces:

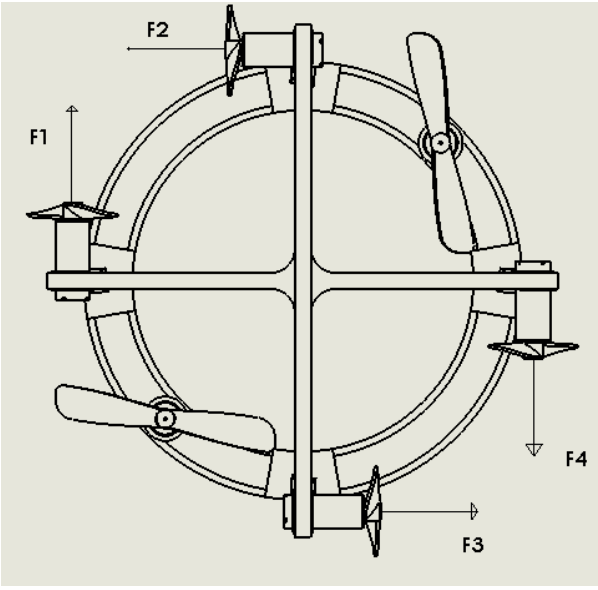
1. Balloon lift
2. Gravity
3. 6 motors applying lift in their respective directions

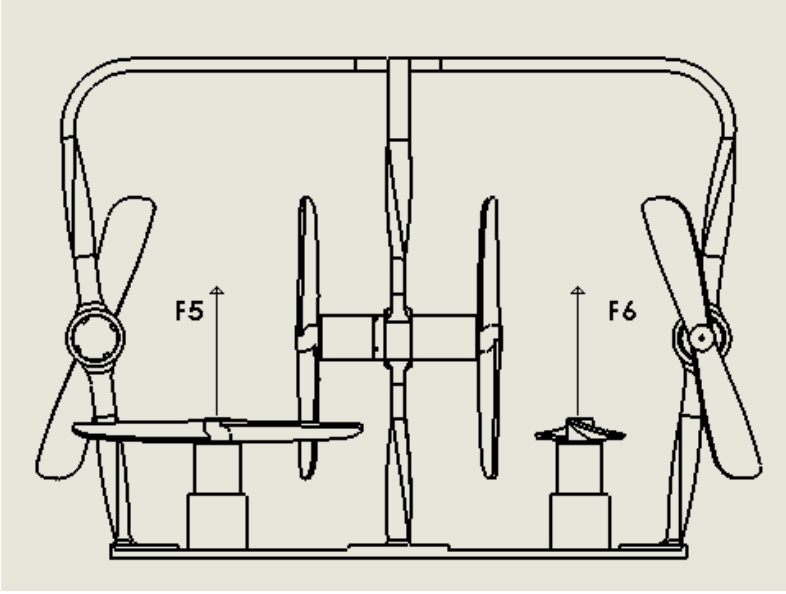
Due to the stable nature of the configuration, roll and pitch control are relinquished.

Lift – vertical lift is applied with motors 5,6 applying force in the upwards direction. Motors 5 and 6 torques cancel out due to symmetry.

Yaw – yaw is controlled using the 4 lateral motors. Motors 2,3 apply an force causing the vehicle to yaw clockwise (respectively motors 1,4). Motors 2,3 rotate in opposite directions in order to cancel-out any torques they apply to the vehicle (although the are very small)

Translation – velocity in the X/Y axis is controlled with the 4 lateral motors. Motors 1,2 and 3,4 respectively apply forces to accelerate in the diagonal direction. Motors 1,3 and 2,4 respectively apply forcers to accelerate in the perpendicular diagonal direction.





Roll and pitching torques produced by motors 1,2,3,4 are neglected because they are relatively weak in comparison to the returning force of gravity. Drag is also present but neglected for the sake of simplicity and since its not in the scope of this work.

The proposed configuration has two downfalls.

a. The configuration requires **6 motors**. The two additional motors entail more electronics and ultimately more weight and an increase in overall complexity.
b. The configuration mixing has a flaw since motors 1-4 downwash dramatically effect motors 5,6 efficiency i.e. there is aerodynamic mixing between motors.

Frame Design, Manufacturing & Assembly

The fame design is driven from the discussed constraints:

1. Weight – minimize!
2. Provide for the assembly of 6 motors in the discussed configuration.

In order to save weight, the new frames structural strength is mainly supplied by the Crazyflie PCB located at the bottom center of the frame. The frame is designed as a one-piece 3D printed from PLA in order to allow quick design iterations, flexiblilty and to save weight.

The following properties were extracted from the Solidworks model of the frame:

Mass = 5.92 grams

Volume = 5915.55 cubic millimeters

Surface area = 10661.92 square millimeters

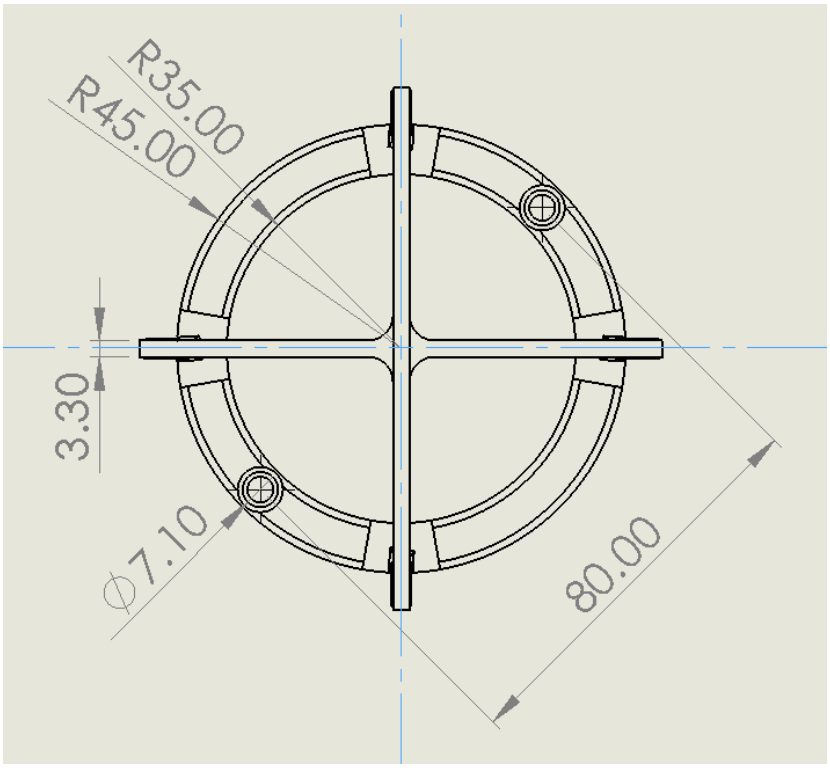
Center of mass: (millimeters)
X = -0.01
Y = 31.38
Z = 0.01

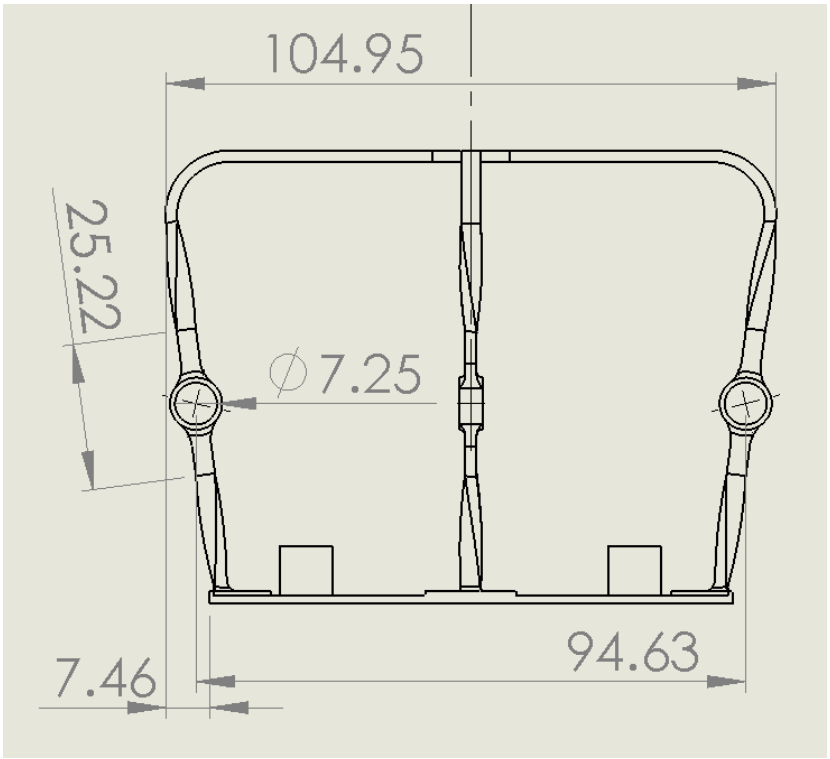
Principal axes of inertia and principal moments of inertia:
Taken at the center of mass.
Ix = (0.00, 1.00, 0.00) Px = 9950.15
Iy = (-0.71, -0.01, 0.71) Py = 10419.42
Iz = (0.71, 0.00, 0.71) Pz = 11298.65

Moments of inertia: (grams * square millimeters)
Taken at the center of mass and aligned with the output coordinate system.
Lxx = 10859.08 Lxy = -2.23
Lxz = -439.62
Lyx = -2.23 Lyy = 9950.17
Lyz = 1.51
Lzx = -439.62 Lzy = 1.51
Lzz = 10858.98

Moments of inertia: (grams * square millimeters)
Taken at the output coordinate system.
Ixx = 16682.34 Ixy = -4.09
Ixz = -439.62
Iyx = -4.09 Iyy = 9950.17
Iyz = 2.84
Izx = -439.62 Izy = 2.84 Izz = 16682.25

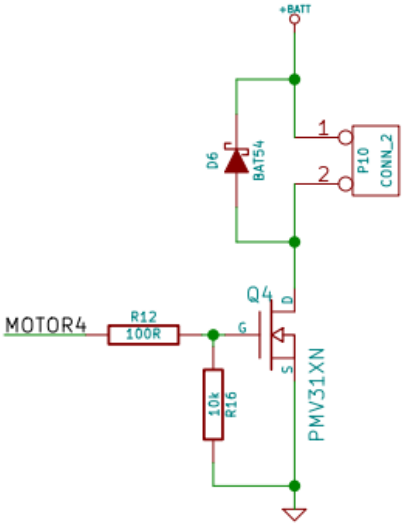
The primary dimensions of the frame are visible in the following drawings [mm]:



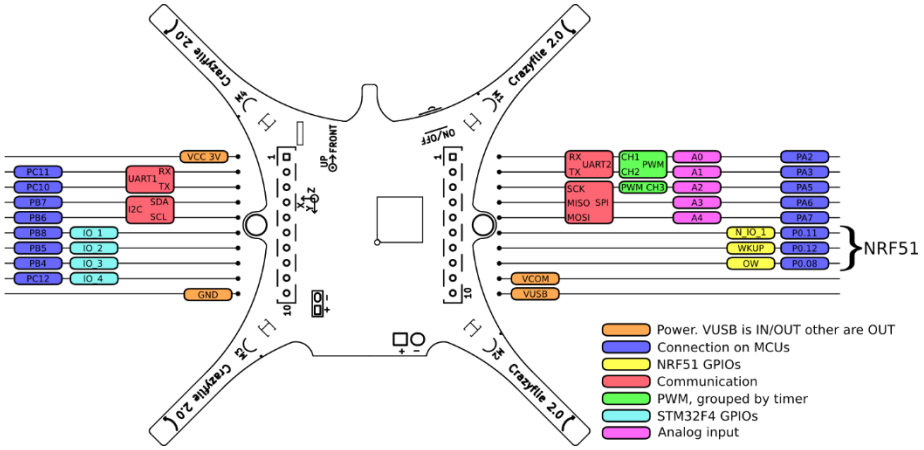


Electronic Design, Manufacturing & Assembly

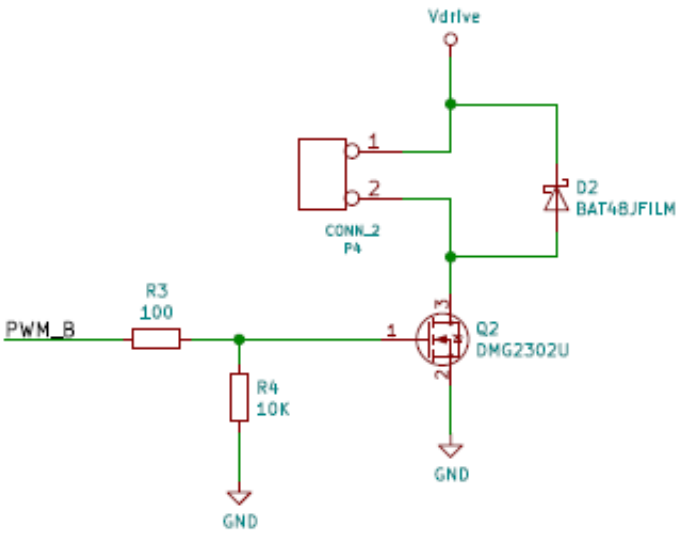
As previously mentioned, in order to control an additional two motors, we need additional electronics. The Crazyflie PCB contains 4 unidirectional DC-motor controllers. The controllers are implemented using PWM signal controlling an N-ch MOSFET switching the motor negative pole, as can be seen in the following excerpt from the schematic:

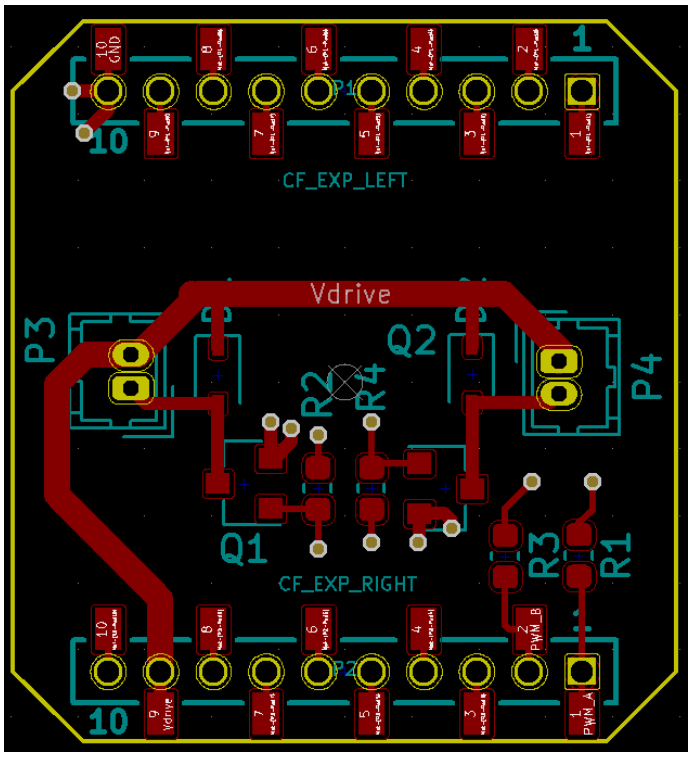


We need to add two more motor controllers of the same fashion. These controllers were implemented on a dedicated expansion board, which is an interface the Crazyflie PCB provides in order to allow extensions. The expansion board design is based on the readily supplied expansion board template provided by bit-craze. The expansion board interface can be seen in the following infographic:



Below is an excerpt of the expansion board schematic and layout:

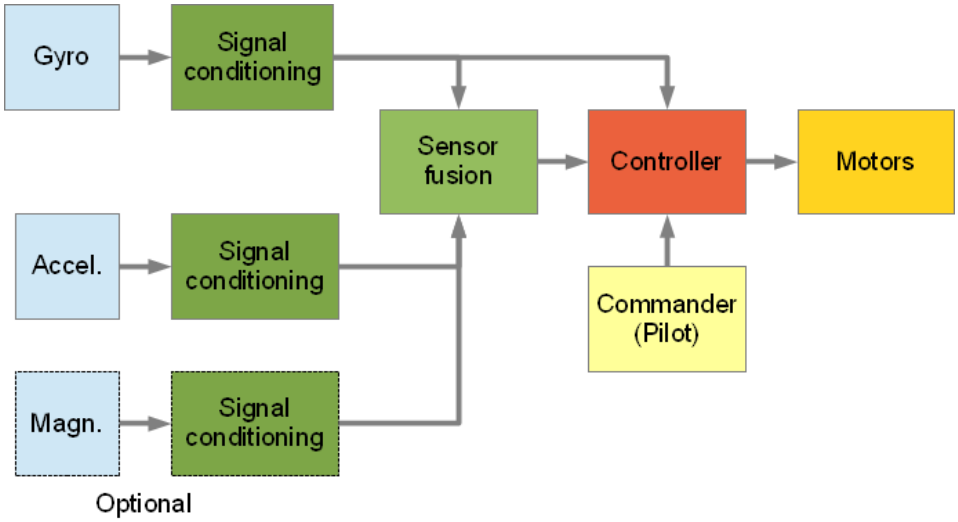


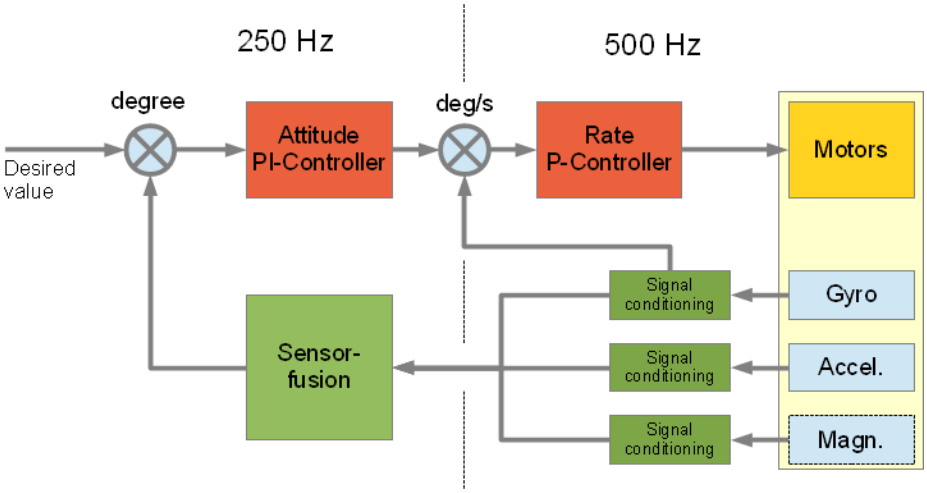


The board was manufactured in china PCB-WAY (2-layer board) and assembled manually upon arrival.

Control Scheme Modifications

The new proposed configuration obviously requires changes in the control scheme and algorithms implemented in the Crazyflie firmware. The original design is depicted in the following two diagrams:

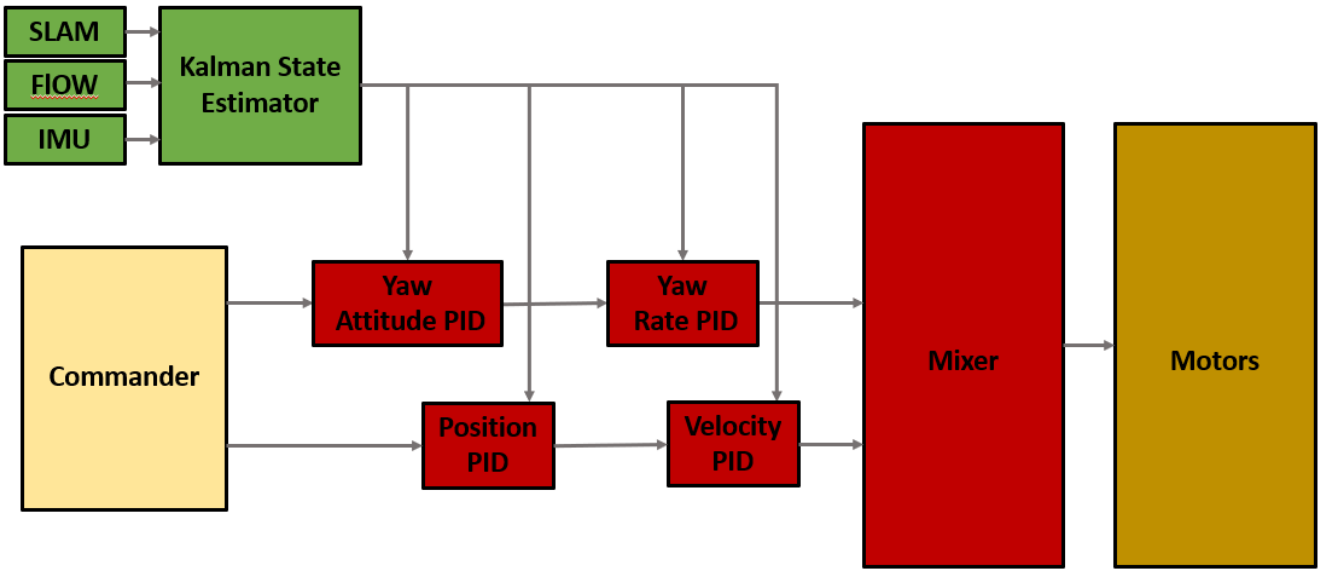




The changes made to the scheme contain the following:

1. Roll and pitch control (attitude + rate) can be relinquished due to natural stability. This greatly reduces the control complexity and demands (nothing occurs on fast time scale).
2. The yaw rate and attitude control loops are left as-is
3. Thrust control remains the same.
4. Position/Velocity controller remains the same. Their output is fed as directly to the mixer
5. The mixer is modified to operate as described previously (this required changes in the motor driver to support an additional two motors using the dedicated expansion deck described)

The new scheme is described in the following diagram:



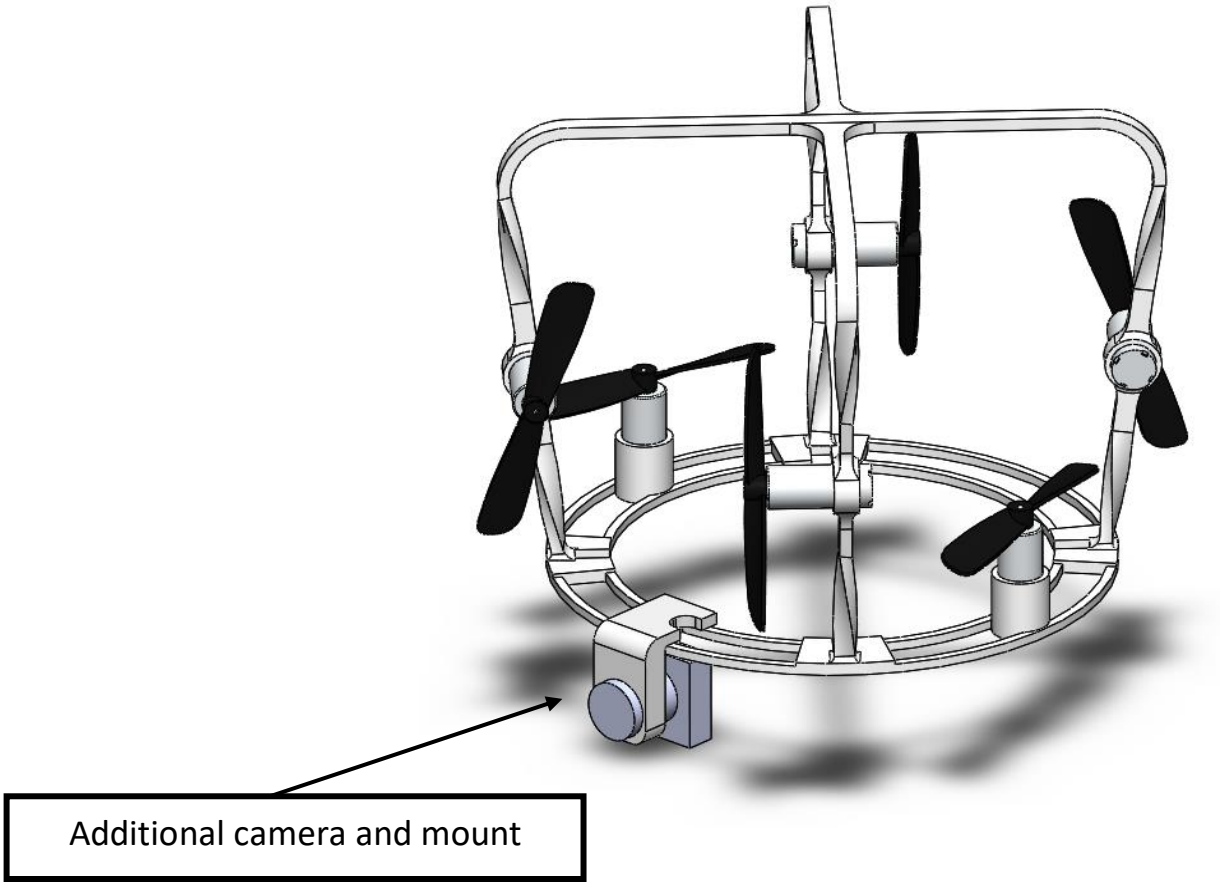
Flight Time - Test Results

Flight tests show that in a hover scenario a flight time of **~25 minutes** can be achieved in this new proposed configuration (with the original battery). This result agrees clearly with simple weight reduction math. The effective weight of the prototype (when placed on a scale whilst attached to the balloon) is roughly **5 grams** which **is six-fold reduction in weight** which translates to a **5 times increase in flight time** (the difference is explained by all the assumptions made along the way)

SLAM Integration

Overview

In order to provide a more accurate position sensor, a micro FPV camera was added to the platform. The camera is used to perform monocular SLAM (offboard – on a nearby PC) and update via radio the Crazyflie with its accurate position in space. The camera is mounted using a dedicated 3D printed adaptor (as can be seen below). These additions obviously add extra weight and reduce flight-time, but this was done as a concept demonstration only.

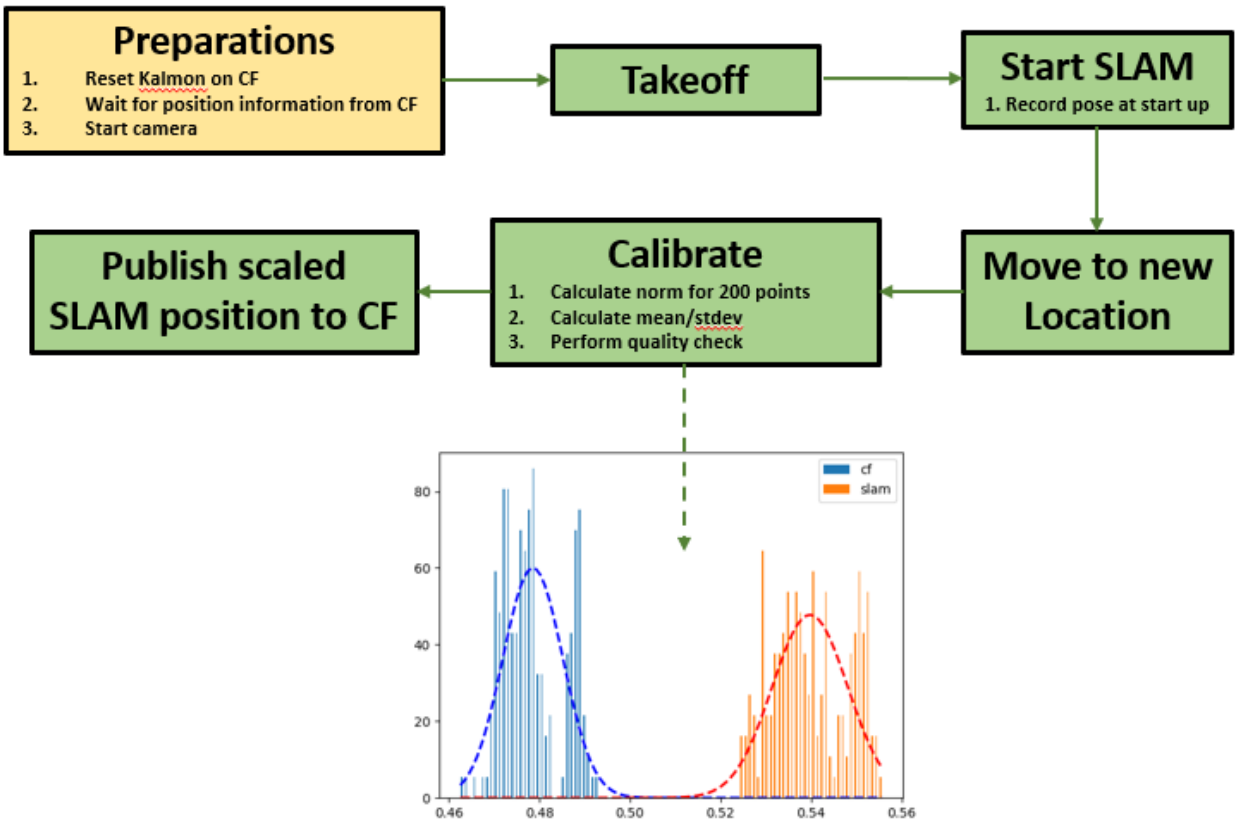


The monocular SLAM algorithm was run in a ROS enabled environment on a PC. A few packages were used for the implementation:

1. **orb_slam2_ros –** a ROS wrapping of the ORB_SLAM2 package. The package subscribes to an image received from the camera and publishes a point cloud and camera pose in the point cloud frame of reference.
2. **usb_cam –** a ROS package that provides the ability to capture video from a device (in our case, the FPV receiver) and publish it as an image topic to ROS
3. **crazyflie_ros –** a ROS package facilitating communication and commands to and from the Crazyflie connected with the Crazyradio to the PC.
4. **TF –** a standard ROS package used for tracking transformations between different frames of reference in time.

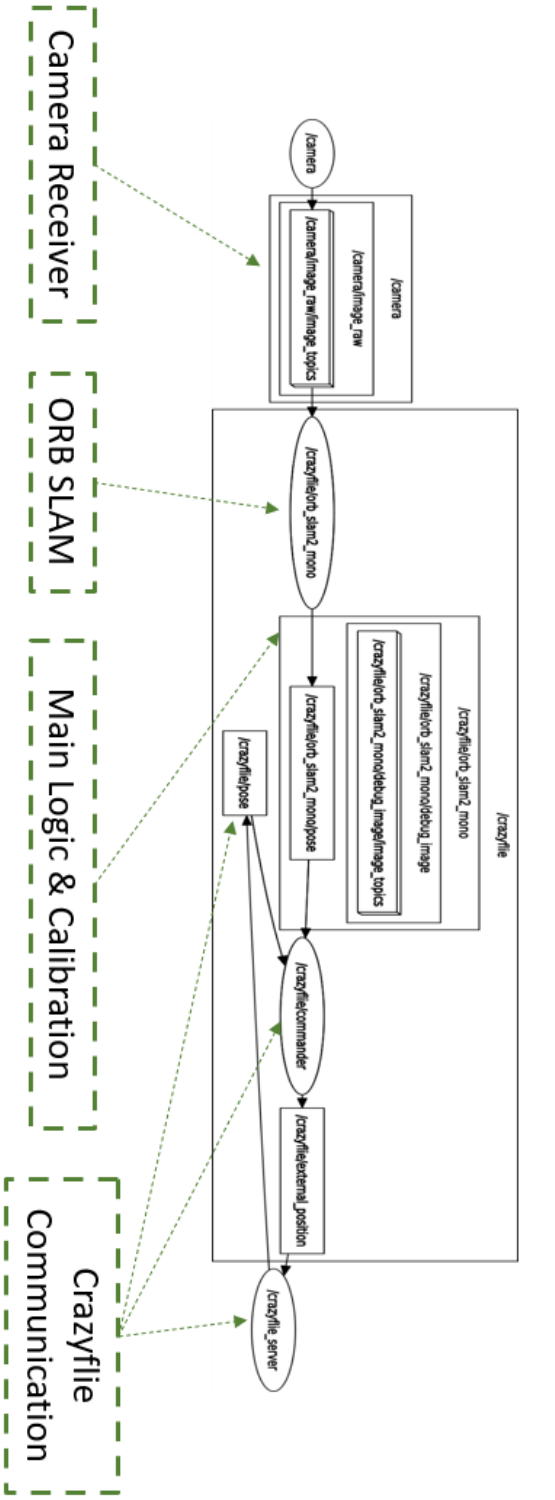
Calibration

Because monocular SLAM cannot determine the scale of the scene a calibration procedure is required in order to calculate the scale estimate. This is done by calculating



the norm of the position in the non-scaled frame and the norm of the position provided by the Crazyflie's own sensors. The norms are calculated for a given fixed position as a mean over several measurements. These norms are then divided to provide a scale estimate. A quality check is then carried out to ensure calibration is adequate.

The following graph was produced with rqt_graph (ROS package) and visualizes the different topics and nodes within the ROS environment running on the PC.



The following tree produced using the TF package describes the frames of reference:

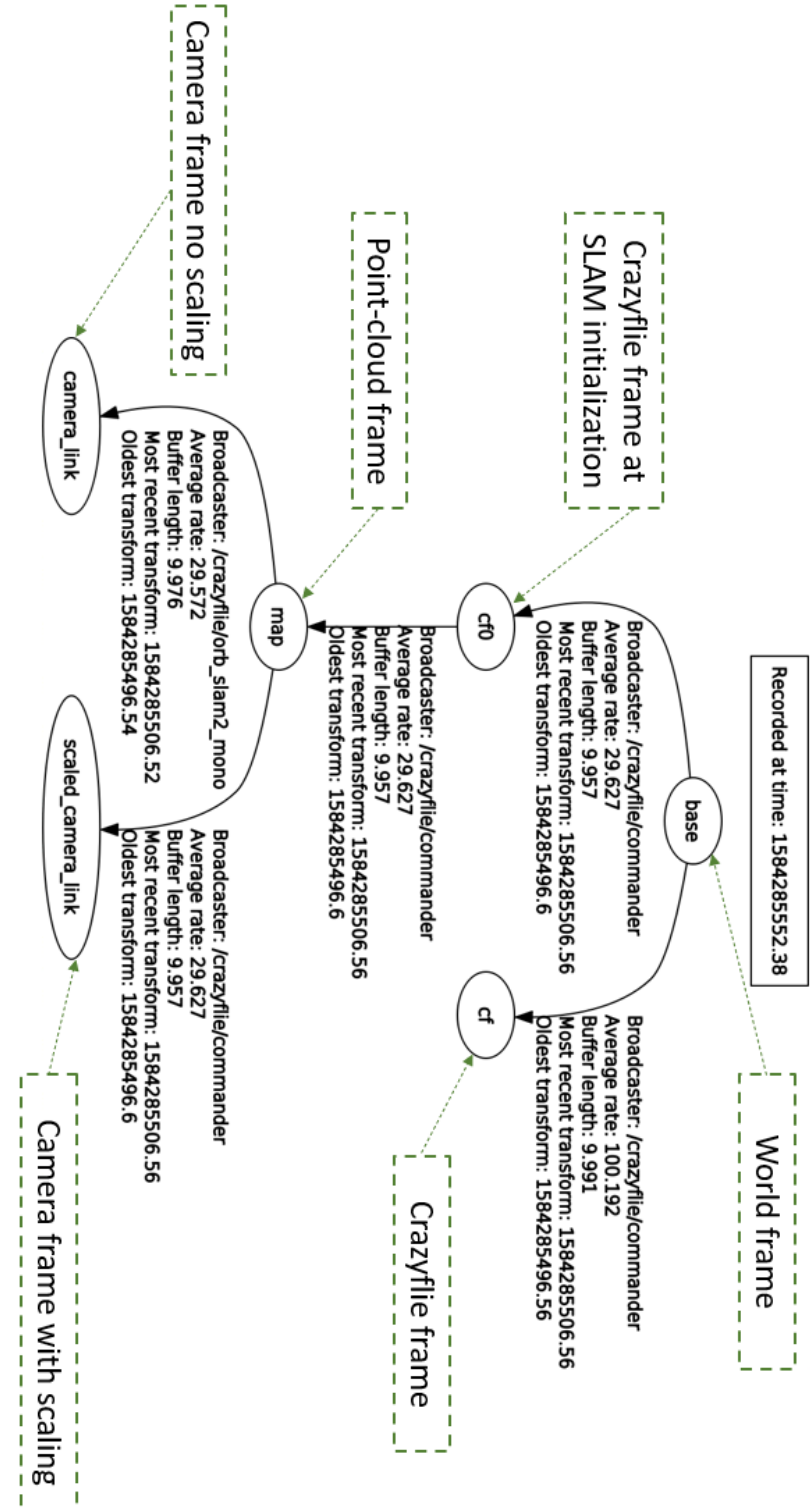


Figure 1

Summary

We set out with the goal of modifying the Crazyflie to boost its flight time without compromising control robustness. This was successfully demonstrated in the project using easy to make/assemble modifications and additions.

We managed to boost the flight time from 5 min to roughly 25 min whilst still providing robust control of position and yaw within a closed space environment. The platform may serve as an easy to use learning and developing platform for future lab students.

We managed to successfully integrate a micro FPV camera onto the platform and demonstrated position control using a SLAM algorithm running online on a nearby PC.

In order to improve these results even further, additional reductions in weight are necessary and if possible, scaling of the balloon. In the future, more improvements can be made to the control algorithm, considering the aerodynamic mixing between the propellers and possibly adding roll/pitch control.

Source Files Repositories

1. Installation & Usage instructions-

   https://github.com/tau-adl/Crazyballoon

2. Crazyflie modified firmware –
   https://github.com/tau-adl/crazyflie-firmware

3. Modified Crazyflie ROS package –
   https://github.com/tau-adl/crazyflie_ros

4. Modified ORB_SLAM2_ROS package -
   https://github.com/tau-adl/orb_slam_2_ros

5. Mechanical and electrical designs-

   https://github.com/tau-adl/crazybal-design-files

References

appliedAI-Initiative. (2020, March 17). *A ROS implementation of ORB_SLAM2*. Retrieved from github: https://github.com/appliedAI-Initiative/orb_slam_2_ros

Bitcraze. (2020, March 17). Retrieved from Bitcraze: https://www.bitcraze.io/

Mur-Artal, R. M. (2015). ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics, 31*, 1147--1163. doi:10.1109/TRO.2015.2463671

ORG, R. (2020, March 17). *Packages for common geometric calculations including the ROS transform library*. Retrieved from github: https://github.com/ros/geometry

ROS ORG. (202, March 17). *A ROS Driver for V4L USB Cameras*. Retrieved from github: https://github.com/ros-drivers/usb_cam

whoenig. (2020, March 17). *ROS Driver for Bitcraze Crazyflie*. Retrieved from github: https://github.com/whoenig/crazyflie_ros