

Team Number: 1732

Advisers: Nicola Elia, Phillip Jones

Team Members/Roles:

Nick Robbins – Project Lead

Cole Beaulieu – Communications Lead

Jake Flores – Webmaster

Grant Manley – Co-Webmaster

Ben Nelson – Concept Leader

Tianxiang Shen

Chengrui Yang

Team Email: swarm\_f17@iastate.edu

Team Website

Contents

[1 Introduction](#_30j0zll)

[1.1 Project statement](#_1fob9te)

[1.2 purpose](#_3znysh7)

[1.3 Goals](#_2et92p0)

[2 Deliverables](#_tyjcwt)

[3 Design](#_3dy6vkm)

[3.1 Previous work/literature](#_1t3h5sf)

[3.2 Proposed System Block diagram](#_4d34og8)

[3.3 Assessment of Proposed methods](#_2s8eyo1)

[3.4 Validation](#_17dp8vu)

[4 Project Requirements/Specifications](#_3rdcrjn)

[4.1 functional](#_26in1rg)

[4.2 Non-functional](#_lnxbz9)

[5 Challenges](#_35nkun2)

[6 Timeline](#_1ksv4uv)

[6.1 First Semester](#_44sinio)

[6.2 Second Semester](#_2jxsxqh)

[7 Conclusions](#_z337ya)

[8 References](#_3j2qqm3)

[9 Appendices](#_4i7ojhp)

# 1 Introduction

## 1.1 Project statement

This is a project revolving around creating a swarm of Crazyflie quadcopters. As it stands currently, the Crazyflie open source platform enables the user to fly one or two crazyflies with one included radio. In order to fly more, we need to change how the radio and ground station process these crazyflies. Once we get more than two flying we are going to also implement crazyflie to crazyflie communication. We also have the benefit of a camera system to tell us exactly where these crazyflies are located.

## 1.2 purpose

The implementations for this project are extremely open. With a swarm of quadcopters being controlled as a group or individually the user could achieve things like carrying a heavy or strangely shaped load, choreographing a multi camera scene, as well as recreational uses like programming them to perform aerial acrobatics.

## 1.3 Goals

We have multiple goals that will build upon each other:

1. Implement addressing so each Crazyflie can be controlled on one channel.
2. Remove the acknowledgement the groundstation waits for.
3. Implement the BigQuad expansion deck to expand crazyflie size.
4. Enable Crazyflie to Crazyflie communication.
5. Be able to lift a load with multiple Crazyflie in sync
6. Control 5-10 Crazyflie on one radio

# 2 Deliverables

We have multiple goals and the deliverables for each will be broken down below:

**Goal 1 deliverable:**

The code that goes along with a video of multiple crazyflies being controlled on one channel.

**Goal 2 deliverable:**

New code base from the groundstation that does not include the ACK and compiles and runs sucessfully

**Goal 3 deliverable:**

A physical working prototype of the Big quad assembled.

**Goal 4 deliverable:**

Code base, as well as a video of Crazyflie to Crazyflie communication, for example one Crazyflie commanding another to do something.

**Goal 5 deliverable:**

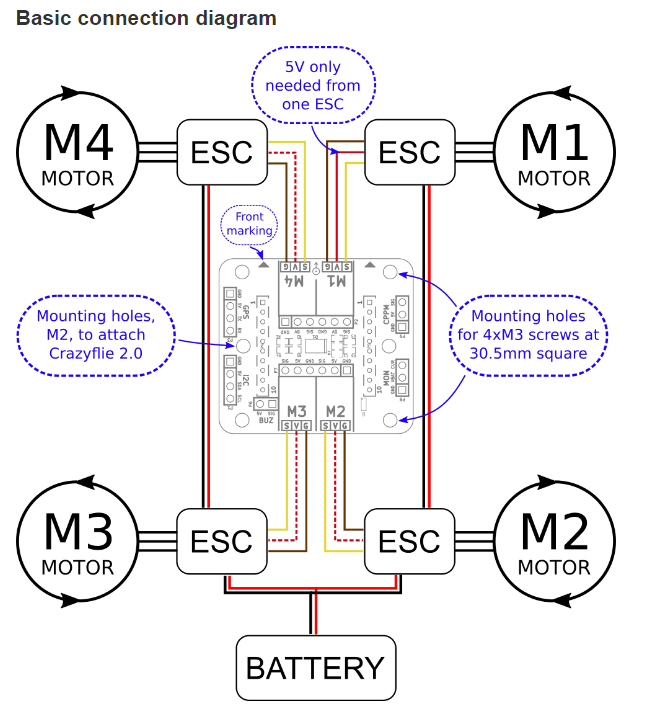
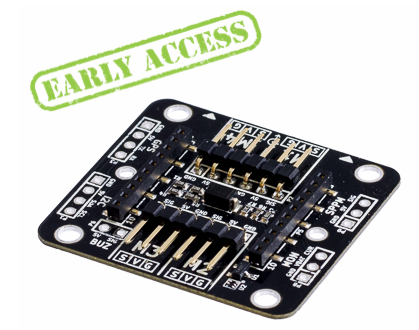
Video of Crazyflies lifting an object in sync

**Goal 6 deliverable:**

A video of a swarm of 5-10 quads on one radio flying together

# 3 Design

The design of this project will be broken up into two groups that will focus on either hardware or software. The hardware group will focus on implementing a current expansion for the crazyflies called a BigQuad board. In order to achieve this, this group will have to select an external frame, four motors, and four ESC’s to connect to the BigQuad board. We will then take the chip that is used to control the crazyflies and connect it to the finished quad.



The second group will work on the software side of the project. As it stands right now the crazyflie emits an acknowledge when it receives the packet from the groundstation. This usually is not a problem. However, when multiple quads are connected there is a hangup that prevents the acknowledged to be read from the groundstation. To solve this problem, we will remove the ack from the communication stack. We will then implement broadcast messaging to increase the speed at which the packets can be sent to the crazyflie. Last, we want to implement Crazyflie to Crazyflie communication. This would essentially remove the need for the groundstation except when commands are being sent.

## 3.1 Previous work/literature

There is a wiki that was developed by a past grad student discussing how he implemented two quads per radio. We will use his groundstation as a baseline for our changes to the communication stack.

## 3.2 Proposed System Block diagram

In our system there will be a Client computer that serves as a ground station to all of the quadcopters that are currently flying. This computer will transmit coordinates on where the quadcopter should fly to via the USB radio. Each of the quadcopters will receive the command via the radio, however only one quadcopter will actually move per radio transmissions. This system will run in a continual loop constantly sending new commands to the quad copters based upon the current position that the quad copters are reporting that they are at.



## 3.3 Assessment of Proposed methods

In order to achieve communication both between the CrazyFlies and the ground station, as well as between the CrazyFlies themselves, there are two main strategies that could be employed:

1. FDMA: Dynamically switch the CrazyFlies’ radio channel to enable communication over multiple frequencies. This would be the simplest solution for reducing interference between CrazyFlies. However, it does not allow for broadcast messages and still introduces interference if two objects attempt to communicate with a third object at the same time on the same channel. Furthermore switching between radio frequencies has proven to be an issue in the past as the time required for the hardware to change channels has cause latency problems. These would need to be overcome if this is to be a viable option for swarm control.
2. Keep all CrazyFlies on the same channel and find another method for multiple access.
   1. TDMA: Time Division Multiple Access calls for giving each radio a time slot to send messages and then wait patiently for their next time slot to send messages again. During down times the object stores messages it desires to send in a priority queue and sends as many as possible during its time slot. The issue with this system is it requires object who may have many messages to send to wait even if other objects that may not have any messages at all. This system would also required well synchronized internal clocks and potentially crashes if any of these clocks get off by milliseconds. The benefits of this system is that it wouldn’t require any alterations to the radio firmware.
   2. CDMA: Code Division Multiple Access is the system implemented by most major wireless providers. In this system all communication is broadcasted across all frequencies and interference is handled by the use of orthogonal spreading codes. This system would need to be paired with a soft ID system if all CrazyFlies needed to be capable of communicating with each other. A downside of this method is that it was developed for a single ground station networking communication between devices and adjustments would need to be made in order to implement it for our purposes. Furthermore, it would require an extensive overhaul of the radio’s firmware.

## 3.4 Validation

We will be able to confirm that our solutions work by validating quad to quad communication and by flying 5-10 quadcopters at one time to perform a currently undefined task using a single radio dongle. The overall success of our project will be directly reflected in the success of this quadcopter cooperation mission.

# 4 Project Requirements/Specifications

## 4.1 functional

1. We will construct a communication platform that will allow the user to control multiple quads simultaneously using a single radio. The total number of quads able to fly at once must be between 5 and 10.
2. We will create a big quad roughly twice the size of the current crazyflies quads we have at our disposal. This quad must have the ability to lift a non-trivial payload.
3. We will develop a system that allows communication between the crazyflies themselves. This system may be required to achieve our primary goal of flying 5 to 10 crazyflies at once, depending on obstacles we encounter during this project.

## 4.2 Non-functional

1. The codebase we construct should be thoroughly documented and designed in such a way as to promote maintainability and extensibility.
2. The codebase should fail gracefully and provide detailed output of any internal errors it encounters
3. The big quad we create should be robust enough to endure minor crashes without sustaining significant damage
4. Groundbase-to-quad and quad-to-quad communication should be fast, ideally with one-way communication times of less than 2ms.

# 5 Challenges

According to the goals we set and the problems we are facing, there are three major challenges we are going to deal with:

* Flying more than 4 quadcopters using one radio
  + communication efficiency among these quads is the point of controlling
  + We will have to come up with a new signal processing technique to avoid interference
  + signal feedback accuracy among quads, ground station and radio is the key to achieve the future loading tasks
* Making quadcopter accomplish picking-up and loading assignments
  + BigQuad feasibility
    - frame selection: majority of frames in market is made by carbon fiber so the material is not a problem but the constructive configuration is needed to be considered
    - motor selection: the key point to determine the motor is the thrust the motor can provide and thrust is most related to the total weight of our entire program. Also, there are more than one way to calculate the thrust which most of them are mathematics. Thus we need to try to find the appropriate thrust.
    - cost limit: need to find more cost efficient parts for quads
  + Quadcopter prototype: we got a previously unused quad model as our prototype drone and we need to implement Crazyflie expansion board on it for testing.
    - Testing motors: because the motors are brushless motors, we need to figure out its working mechanism and run them. By now, after trying many methods, we still can’t run them.
    - External ESC: need to be familiar with the ESC-motor system and manipulate it properly.
* Controlling quads to achieve tasks
  + design a installation to pick up the payload
  + guarantee the load stabilization during lifting process
  + unleash the load quickly and nondestructively

# 6 Timeline

**SEE GANTT CHART ATTACHED**

# 7 Conclusions

In conclusion, at the end of this project we want to control multiple Crazyflies and let them cooperate with each other to fulfill a task like carrying a load. We will also build an expansion deck to expand the size and payload of the Crazyflie. In order to achieve these goals, we will expand the communications suite so each Crazyflie can be controlled on one channel, and then control multiple Crazyflies on one radio frequency bandwith. To make Crazyflies work in sync, we will enable Crazyflie to Crazyflie communication.

# 8 References

<https://wikis.ece.iastate.edu/distributed-autonomous-and-networked-control-lab/index.php/Distributed_Autonomous_and_Networked_Control_Lab_Wiki>

<https://wiki.bitcraze.io/projects:crazyflie2:expansionboards:bigquad>

# 9 Appendices

