Quadcopter Swarm Research

Project plan

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 - Co-Idea Leader

Chengrui Yang - Co-Idea Leader

Team Email: swarm\_f17@iastate.edu

Team Website: may1732.sd.ece.iastate.edu

Contents

[1 Introduction………………………………………………………………………………………………………… 2](#_Toc442284807)

[1.1 Project statement………………………………………………………………………………………………..2](#_Toc442284807)

[1.2 purpose…………………………………………………………………………………………………………….. 2](#_Toc442284807)

[1.3 Goals……………………………………………………………………………………………………………….... 2](#_Toc442284807)

[2 Deliverables…………………………………………………………………………………………………………. 2](#_Toc442284807)

[3 Design…………………………………………………………………………………………………………………. 3](#_Toc442284807)

[3.1 Previous work/literature………………………………………………………………………………….... 4](#_Toc442284807)

[3.2 Proposed System Block diagram………………………………………………………………………...](#_Toc442284807)5

3.3 Assessment of Proposed methods………………………………………………………………………5

3.4 Validation………………………………….………………………………………………………………………7

[4 Project Requirements/Specifications……………………………………………………………………. 7](#_Toc442284807)

[4.1 functional…………………………………………………………………………………………………………..7](#_Toc442284807)

[4.2 Non-functional………………………………………………………………………………………………...](#_Toc442284807)..7

[5 Challenges…………………………………………………………………………………………………………...8](#_Toc442284807)

[6 Timeline……………………………………………………………………………………………………………... 9](#_Toc442284807)

[7 Conclusions…………………………………………………………………………………………………………. 9](#_Toc442284807)

[8 References………………………………………………………………………………………………………….. 9](#_Toc442284807)

[9 Appendices………………………………………………………………………………………………………..10](#_Toc442284807)

# 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. Create mesh networking firmware on ESP8266 chip
2. Enable crazyflie to accept data from wifi module through UART port
3. Remove the acknowledgement the groundstation waits for.
4. Implement the BigQuad expansion deck to expand crazyflie size.
5. Enable Crazyflie to Crazyflie communication.
6. Be able to lift a load with multiple Crazyflie in sync
7. Control 5-10 Crazyflie at one time

# 2 Deliverables

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

**Goal 1 deliverable:**

A test that shows we can send both broadcast messaging and single packets to any wifi module.

**Goal 2 deliverable:**

Be able to send a packet to the wifi module connected by the crazyflie’s UART port and have the crazyflie accept the data.

**Goal 3 deliverable:**

Run the ground station code without the acknowledge and have it work the same as before.

**Goal 4 deliverable:**

A working big quad copter using the crazyflie controller.

**Goal 5 deliverable:**

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

**Goal 6 deliverable:**

Video of Crazyflies lifting an object in sync

**Goal 7 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. The group has currently selected an external frame, four motors, and four ESC’s to connect to the BigQuad board. At the current moment this team is trying to figure out how current quadcopters (that are in the lab) are built in order to their own quadcopter from the current parts set



The second group will work on the software side of the project.This team is trying to implement a mesh network between all of the quad copters, this will allow all of the quad copters to be able to talk to any other quad copter in live time. The end goal for this would create a system that would essentially remove the need for the ground station 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 ground station as a baseline for our changes to the communication stack.

Some other institutes also have succeeded in making a multiple quadcopter system ( M.I.T. and University of Pennsylvania ) however most of their designs and work are proprietary and not available to the public. Their designs for their systems are very similar to ours in that they have a camera system to look at all of the quads in the set area and they have a ground station to communicate with all of the quads at once. The quadcopters that MIT uses is a custom build small quad while University of Pennsylvania uses the same crazyflie quadcopter base that we are using. Research that is published from both of these universities will come in helpful as we continually fully develop our systems.

## 3.2 Proposed System Block diagram



 Above is our proposed system block diagram. The camera system will relay position information on the quadcopters to the ground stations, which will interpret the position data and make control decisions for the quadcopters. Most of this system is already implemented. The ground station will then create the CRTP control packets and send them to the attached WiFi module over a serial connection.

 The WiFi module will then broadcast that data over UDP to a WiFi module on the crazyflie itself, which will relay the control packet to the microcontroller on the crazyflie.

## 3.3 Assessment of Proposed methods

In order to achieve communication between both the CrazyFlies and the ground station and between the CrazyFlies themselves, our team explored several forms of multiple access:

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, because the time required for the hardware to change channels causes latency problems. These would need to be overcome if this is to be a viable option for swarm control.
2. TDMA: Time Division Multiple Access calls for giving each radio a time slot to send messages and then wait in a listening state 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 then sends as many of these messages as possible during its transmission slot. The issue with this system is that it requires an object that may have many messages to send to wait for sending even if no other objects have messages to transmit. This system would also required well synchronized internal clocks and potentially crashes if any of these clocks go out of synch. The benefits of this system is that it wouldn’t require any alterations to the radio firmware.
3. 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. The 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 is the most complex form of multiple access to implement and would require an extensive overhaul of the radio’s firmware.

After exploring all of these options for implementing our requirements in our current communication system, we decided to create a completely new system. Although this choice created more overhead, it had the benefit of being designed down to the hardware with our team’s needs in mind; a much less hacky way to start implementation which we hope will lead to easier scaling later on. This new system will utilize ESP2688 WiFi modules to facilitate communication. These modules were chosen because they had been used successfully in our lab previously and met all of our requirements: UDP messaging, broadcast messaging, point to point communication, and low latency according to preliminary testing. Further proof-of-concept trials have since been run and these functionality requirements have all been proven. The choice to replace the CrazyFlie Radio dongle with the ESP2688 WiFi module is the largest design decision we have made this semester.

Notes on other design decisions:

UDP vs TCP: Our team chose to replace the TCP style communication protocol originally in place with a new UDP system. UDP is favorable for our needs because most of the data our system sends is location data delivered to the ground station via the camera system. With this kind of data, message integrity is not a priority of ours, because it is always more desirable to send the newest location data rather than resend old data.

Remainder of communication stack: Our team has focused on replacing the transmission and reception aspects of the communication stack, but for the most part the remainder of the communication stack is still the original CrazyFlie firmware. For simplicity, we are using the ESP2688 modules to wrap the packets in the format expected by the Syslink firmware before forwarding them to the uart port of the CrazyFlie. While this is not the most efficient process and we plan to eventually replace the current firmware, we deemed it out of the scope of this design iteration and will be implemented in the Spring semester.

## 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 wifi modules in place of the onboard radio. 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 multiple wifi modules. 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 wifi modules in place of the current radio
	+ communication efficiency among these quads is the point of controlling
	+ We will have to design a mesh networking system between the quads
	+ We will have to rewrite the firmware to accept the new wifi module in place of the onboard radio.
* 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 using a wifi module instead of the onboard radio which will allow us to expand our signal size significantly. As an added feature, we will enable Crazyflie to Crazyflie communication in order to expand on their communication capabilities.

# 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](https://wiki.bitcraze.io/projects%3Acrazyflie2%3Aexpansionboards%3Abigquad)

<https://github.com/olab-io/meshish>

<http://www.whatimade.today/esp8266-easiest-way-to-program-so-far/>

<https://github.com/bitcraze/crazyflie-firmware>

<https://github.com/whoenig/crazyflie_ros>

# 9 Appendices

