ECE 453 Project Proposal (Fall 2018)

University of Wisconsin-Madison
Vaughn Kottler, Mayank Katwal, Cooper Green

1 Introduction

We are interested in building a *quadcopter* plus *ground station* and *web-based user interface*. We have chosen to call this project the **fault-tolerant quadcopter**. This name reveals one of our design goals that will be covered in a future section.

This document serves as the formal proposal to be vetted by the course instructor, Joe Krachey. We have additional documentation in work online that we plan to keep in sync with our project's scope and current progress. At the time of writing it is not yet in a stable state.

This project is designed for three major bodies of work that were mentioned above but are better captured by **Figure 1**:

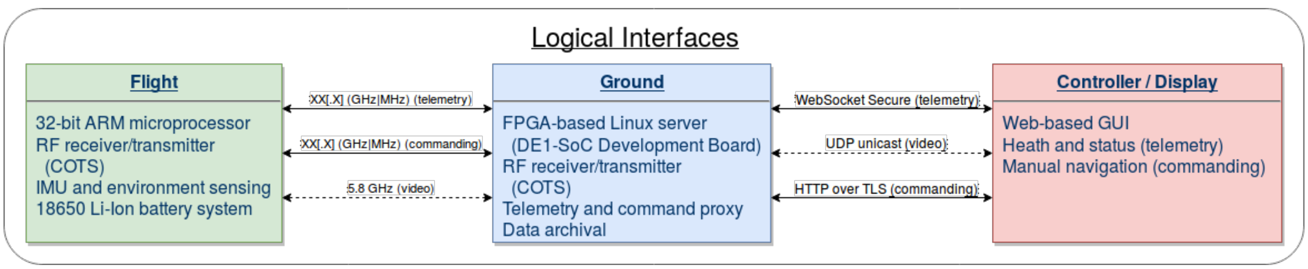

Figure 1: High-level overview of the major components and their interfaces

This high-level architecture is inspired by existing aerospace avionics and software systems that we have done research on and have some first-hand experience with. Our current, collective experience with such systems (and the technical challenges we anticipate being associated with them) is minimal, though. For this reason **we seek feedback on our lower-level goals and approach**, provided that this high-level idea suffices as a project worth pursuing.

2 Technical Features

A vehicle that is *single-fault tolerant* is capable of continuing nominal operation after experiencing any arbitrary failure in a well-defined *fault space*. We aim to implement single-fault tolerance by:

1. Limiting the initial fault space to a "loss of ground station heartbeat" event
2. Executing a "landing maneuver" upon fault detection
3. Iteratively hardening our design to a broader fault space, time permitting

We recognize some intermediate milestones that will need to be reached before an "automatic flight-termination system" described above can be expected to function:

- Establish wireless communication between the vehicle and ground station
- Establish percentage-based throttle control over each motor
- Establish manual-commanding capability to the vehicle from a web-based user interface
- View live telemetry from a web-based user interface
- Sense angular velocity via gyroscope and force experienced via inertial measurement unit

- Develop a control algorithm to fly in a stable hover or holding pattern
- Extend control algorithm to control for velocity in three axes to achieve controlled motion
- Sense relative altitude
- Extend control algorithm to control for a specific *delta-y* (perpendicular to ground plane)

Our limited understanding of control theory and lack of experience in general with avionics systems may limit what we can achieve in the end, but we recognize this and have focused on primarily *system-level* goals and requirements versus specific technical requirements (battery life, thrust and payload capability, etc.).

2.1 Quadcopter

We intend to pursue an implmentation depicted in **Figure 2**:

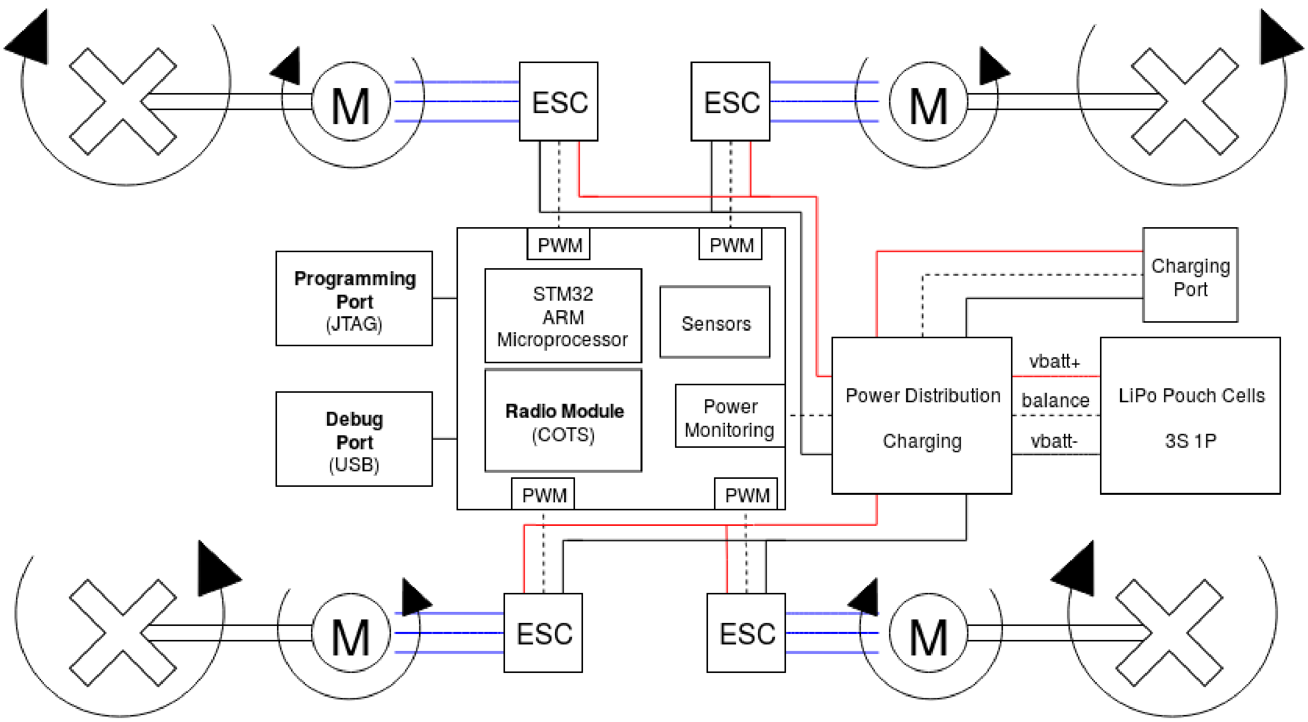

Figure 2: Block diagram view of the quadcopter

Responsible Engineer: **Vaughn**

2.2 Ground Station

We intend to pursue an implmentation depicted in **Figure 3**:

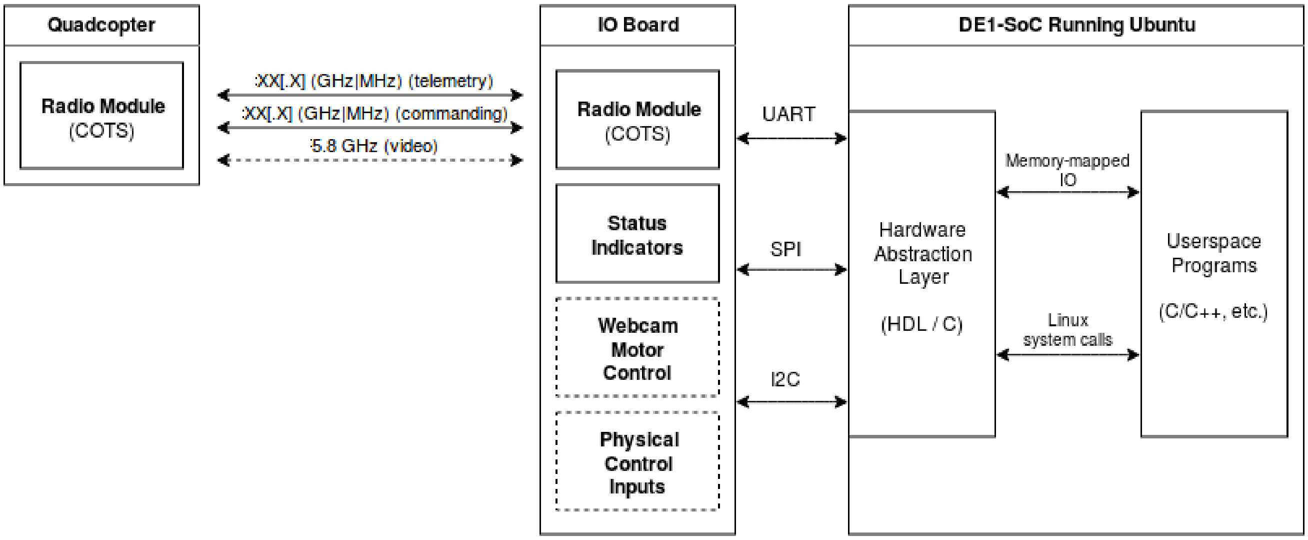

Figure 3: Block diagram view of the ground station

Responsible Engineer: **Cooper**

2.3 Display and Controller

We intend to pursue an implmentation depicted in **Figure 4**:

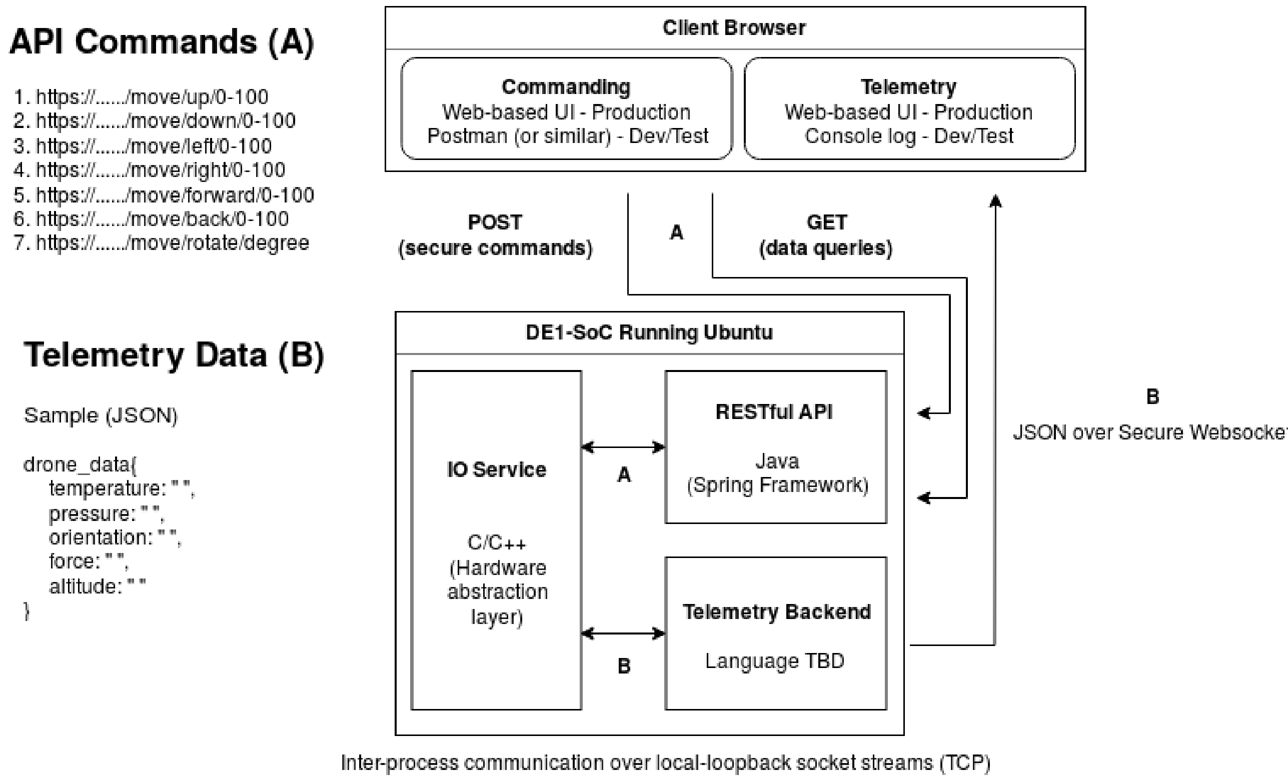

Figure 4: Block diagram view of the display and control user interface

Responsible Engineer: **Mayank**

3 Roles and Responsibilities

How we plan to share responsibility.

3.1 Vaughn Kottler

Responsible Engineer for the following:

Flight Vehicle TODO

Control Algorithms TODO

3.2 Mayank Katwal

Responsible Engineer for the following:

Telemetry Display TODO

Vehicle Controller Hardware of Software

3.3 Cooper Green

Responsible Engineer for the following:

Ground Station TODO

Radio Frequency Communication TODO

4 Project Management

> *"If I had an hour to solve a problem I'd spend 55 minutes thinking about the problem and 5 minutes thinking about solutions." - Albert Einstein*

TODO

4.1 Initial Development and Prototyping

TODO

4.2 Final Stages

TODO