Software/Hardware Requirements Specification

for

Anzhelka

Version 1.0 approved

Prepared by
Luke De Ruyter
Cody Lewis

University of California, Riverside

May 4, 2012

Table of Contents

Revisions

Current project status and files can be found at

blog.anzhelka.com
code.anzhelka.com

Version	**Date**	**Changes**	**Commiter**
0.01	April 30, 2012	Initial layout of file was created.	Cody
0.02	May 4, 2012	Purpose, Audience, Scope, Perspective, Functions, and Users was populated.	Cody, Luke

1 Introduction

1.1 Purpose

Anzhelka is a complete system intended for autonomous quadrotor flight. Included as a part of Anzhelka is both hardware and software. This includes the quadrotor frame, control electronics, ground station software, and the complete system documentation. Anzhelka is completely open source, and all project files are available for download. You can find in this document any instructions necessary for understanding the functionality of Anzhelka components. This includes hardware and software interfaces, features, and system requirements.

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

This document is written for Anzhelka developers. This document is intended to refine development direction, and to bring new developers up to speed. For this document, developers include software writers, hardware designers, and system testers. When the word ”you” is written, it is referring to the developer who is reading this document. When the words ”the user” is written, it is referring to the end user of the Anzhelka platform.

You should read this document based on your background with Anzhelka. Current developers can find the appropriate section to read. New developers should read the introduction, overall description, and system features sections. If you are a non-developer for this project, and don't intend to ever become one, you should avoid this document. Look on the Anzhelka website for something more appropriate to your needs.

1.4 Product Scope

Anzhelka consists of four main components: a quadrotor hardware frame, custom quadrotor software, ground station software, and detailed documentation via the Anzhelka website. Even without a degree in control systems, you can use Anzhelka components to make an autonomous quadrotor system. By using these components you can customize the functionality of the system to suit your needs, or use them directly to perform predefined commands.

1.5 References

2 Overall Description

2.1 Product Perspective

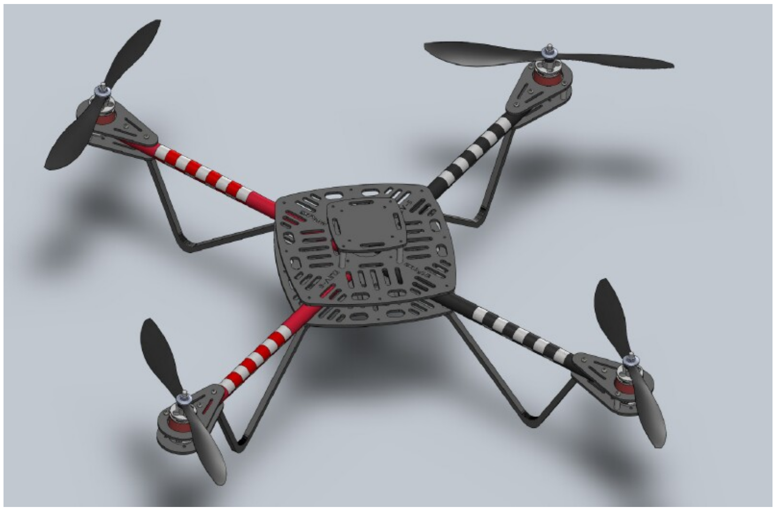

Figure 1: This is an artistic rendering of the frame used for Anzhelka.

Anzhelka is a platform for experimentation in aerial robotics. It is an independent project designed from the ground up. We are taking an existing open hardware frame as our base hardware platform. We have chosen to build on top of this hardware because of its expandability and its durability. We have designed a custom control board that will be able to run all of Anzhelka's software and house all the necessary components.

2.2 Product Functions

Anzhelka is designed to be programmatically controllable in a number of different aspects. The quadrotor can receive commands to set any of it's internal registers, which then affect the flight of the quadrotor. In addition, the quadrotor has a command interpreter that is able to read command sequences from onboard memory. With these commands, the user can control a number of quadrotor functions including attitude and position.

Anzhelka is equipped with a Parallax Propeller multicore processor. The quadrotor has two separate batteries. One battery provides power for the motors and servos; the other battery powers the electronics. The electronic hardware is designed with expandability in mind. The motors and servos have voltage and current monitoring. The Anzhelka system is designed to allow for the easy attachment of additional electronics.

2.3 User Classes and Characteristics

2.3.1 User Example: Jason

Figure 2: Jason Lariart

Age: 28
Education: Some College
Marital Status: Open
Occupation: Videographer
Hobbies: Photography, Videography, Art, Protesting

About Jason: He is an avid videographer and likes to push the limits of his skills, crew, and equipment. He likes to take pictures and videos of things that not just any ordinary person can do. Jason is currently an amateur photographer, but wants to make a professional career out of his work.

Scenario: Jason is preparing for a remote video shoot and realized that he won't be able to get his aerial shot because a helicopter is out of his budget. Jason told the client that he was going to have to spend more money if wanted to get the aerial shot as planned. The client was upset to hear about this and had no choice but to drop the scene from the shoot.

2.3.2 User Example: Dan

Figure 3: Dan Knuth

Age: 82
Education: Doctorate in Mathematics
Marital Status: Widowed
Occupation: Software Developer
Hobbies: Open source programming, online forum posting, sitting

About Dan: Dan works on open source systems. He has literally been around since the begining, and he remembers the days of punch card programming. Throughout his career he has been a strong supporter of making computer science more accessible, and his work in the open source field is both notable and well respected.

Scenario: Dan wants to spend more time focusing on his hobbies, but he doesn't have much time left and he needs to eliminate routine activities. Dan wants to develop a robot to automatically pick up his medicine from the pharmacy and carry it to him. This robot needs to be able to fly, and to be user programmable. Dan has lots of experience in working with software, so it's ok if the product is a bit rough and requires the use of a command line. Dan would be particularly appreciative if the instructions could be programmed in by a series of toggle switches and incandescent lights.

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

3 External Interface Requirements

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software or Hardware Interfaces

3.4 Communications Interfaces

4 System Features

4.1 System Feature 1

4.1.1 Description and Priority

4.1.2 Stimulus/Response Sequences

4.1.3 Functional Requirements

4.2 System Feature 2 (and so on)

5 Other Nonfunctional Requirements

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software or Hardware Quality Attributes

5.5 Business Rules

6 Other Requirements

A Glossary

B Analysis Models

C To Be Determined List