浙 江 大 學

本 科 生 毕 业 设 计 报 告

项目名称 基于 WiFi 通信的四旋翼飞行器遥控系统设计

姓　　名 宋博

学　　号 3110101698

指导教师 卜佳俊

专　　业 计算机科学与技术

学　　院 计算机学院

A Dissertation Submitted to Zhejiang University for the Degree of Bachelor of Engineering

TITLE: WiFi Remote Control System Design for Quadrotor

Author: Song, Bo

StudentID 3110101698

Mentor Bu, Jiajun

Major: Computer Science&Technology

College: Computer Science

Submitted Date: 6/3/2015

浙江大学本科生毕业论文（设计）诚信承诺书

1.本人郑重地承诺所呈交的毕业论文（设计），是在指导教师的指导下严格按照学校和学院有关规定完成的。

2.本人在毕业论文（设计）中引用他人的观点和参考资料均加以注释和说明。

3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。

4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

毕业论文（设计）作者签名：

2015 年 6 月 3 日

摘要

四旋翼飞行器，又称四轴飞行器，是依靠四个螺旋桨的高速旋转来产生升力的无人机。近年来得益于微机电系统(MEMS)的发展，四旋翼飞行器的控制技术已经趋于成熟。本论文的研究重点在于四旋翼飞行器 WiFi 遥控系统的设计，通过手机平台实现对四旋翼飞行器的便捷遥控。

本文先介绍四旋翼飞行器的发展现状，然后介绍并分析现行的基于 2.4GHzPPM 编码的遥控方案的优缺点。从而引出 WiFi 遥控飞行器的优势及其可行性，并详细介绍 WiFi 遥控系统的技术细节。

论文最后对设计成果进行了试飞测试和功能扩展：结果表明本文设计的基于 WiFi 通信的四旋翼飞行器遥控系统能够可靠、高效地完成对四旋翼飞行器的控制任务。且该系统具有高扩展性，进一步开发了手机姿态遥控和航拍功能。

关键词 四旋翼飞行器、Multiwii、航拍、Openwrt、安卓开发

Abstract

Quadrotor, also known as quadcopter, is a kind of UAV lifted by the high speed spin of its four propellers. Thanks to the development of MEMS , the control technology of the quadrotor has become mature. This paper focus on the design of a WiFi remote control system for quadrotor, specifically, developing an Android APP to control quadrotor via WiFi conveniently.

This paper provides the state of the art introduction to the quadrotor technology, and then introduces and comments on the advantages and disadvantages of the current remote control scheme based on 2.4GHz PPM coding. This leads to the need of WiFi remote control system and the elaboration of the technical details of it.

Finally, flight-test and function extensions are conducted. The result shows that the system designed in this paper can control the quadrotor reliably and conveniently. The system is also highly extendable. It can be modified to support phone attitude control and aerial photography.

Keywords quadrotor, Multiwii, Aerial photography, Openwrt, Android

目录

第1章 项目背景

1.1 四旋翼飞行器简介

四旋翼飞行器，又称四轴飞行器，是依靠四个螺旋桨的高速旋转来提供升力的结构简单飞行器。早在 1907 年，法国的 Breguet-Richet 就设计并且成功试飞了世界上第一个四旋翼飞行器“Gyroplane No. 1”[1]。但由于四旋翼飞行器的控制难度较大，早期的技术水平无法实现飞行器的自主飞行控制，大型四旋翼飞行器的发展一直都比较缓慢。近年来得益于微机电控制技术的发展，四旋翼飞行器的控制技术已经趋于成熟[2]。以四旋翼飞行器为载体，其上搭载摄像头等传感器硬件，其内运行计算机视觉等高级算法程序的研究受到广泛的关注[3]。四旋翼飞行器在航拍、勘测等领域也有广阔的应用前景。

1.2 四旋翼飞行器的发展现状与不足

随着四旋翼飞行器的成本和技术难度不断降低，它吸引了越来越多的航模爱好者的关注，各种从事四旋翼飞行器开发的商业公司也如雨后春笋般出现。

大疆创新科技公司成立于 2006 年。它主要从事无人机控制系统和航拍解决方案的研发。它的产品原型来自于一位香港科技大学电子工程系本科生的毕业设计。目前已占据全球 70%的小型无人机市场。

Parrot 公司成立于 1994 年，总部设在法国巴黎，是除了大疆以外的拥有市场份额最大的公司。它于 2010 年推出了可用 iPhone 控制的四旋翼小型无人机。

3D Robotics 是北美最大的面向个人用户的无人机厂商。不同于大疆封闭的代码管理机制，3D Robotics 的代码全部开源，它和大疆的关系正如苹果的 iOS 和 Google 的 Android。

在航模爱好者领域，爱好者们也开发出了诸多优质的四旋翼无人机开源项目。比较著名的有 Arducopter，Openpilot，Multiwii，Pixhawk 等。

无论在商用领域还是民用爱好者领域，四旋翼飞行器大多是由 2.4GHzPPM 调制的无线电信号进行遥控，并应用跳频技术增强信号的抗干扰能力[4]。应用 PPM 编码和跳频技术的 2.4GHz 遥控器具有抗干扰能力强，传输距离远，实时性好，可靠性高等优势。但 2.4GHz 遥控器的价格偏高，体积较大，不便于携带。

并且，由于 PPM 的编码方式的过于简单，带宽过低，无法用它实时回传航拍图像。

图 1.1 2.4 GHzPPM 遥控器

第2章 项目实施方案

2.1 系统需求分析

WiFi 四旋翼飞行器主要由四个部件构成，各个部件的需求分析如下。

2.1.1 传统四旋翼飞行器

传统四旋翼飞行器是整个系统的基础。该项目先搭建一个传统的基于 2.4GHzPPM 遥控的四旋翼飞行器，然后在该飞行器的基础上进行模块的增添和修改。所以要求该四旋翼飞行器能够平稳可靠地完成各种飞行动作，包括升降、俯仰、横滚和偏航。

此外，因为该项目要对原有的飞行器部件进行改造，所以此处应使用扩展性好的，便于组装的硬件和开源的软件。

最后，飞行器应该具备额外的负载能力，能够把 WiFi-串行数据转换模块和图像回传模块(摄像头)载起并平稳飞行。

2.1.2 WiFi 模块

当前主流的飞控板并没有搭载 WiFi 模块，而是用 6 个左右的 GPIO 端口与 2.4GHz 的接收机相连，用来获取遥控器的遥控信号和指定的飞行模式。此外，一些飞控板还支持串口遥控的协议，预留了一个串口作为备用的遥控输入。为了实现通过手机的 WiFi 信号控制无人机，我们需要一个 WiFi-串行数据转换模块，连接到飞控板的串口上。用来实现 WiFi 与串行数据的相互转换。

该模块需要具有高可靠性，低耗电性以及较轻的质量。高可靠性保证该模块可以稳定地进行信号转换，不可以突然当机，否则无人机在高空中会发生不可预知的危险；低耗电性可以提高无人机的续航时间，工作电压最好与板载电压一致，为 3.3V 或 5V；较轻的质量同样可以提升无人机的续航时间，且不会对原无人机的重心产生较大影响。

2.1.3 手机端遥控 APP

手机端的遥控 APP 应该使用 Android 或 Objective-C 编写。该 APP 需要具有友好的用户界面，便于用户控制无人机；需要将遥控数据按照指定的数据协议

打包，通过 WiFi 发送；需要能接收 WiFi 信号，显示飞行参数。且该 APP 同样需要具有高可靠性和快速响应能力。如果延迟过大会导致飞行器控制不稳，容易引发事故。

2.1.4 图像实时回传系统

图像实时回传系统需要将机载摄像头捕捉到的画面实时传回地面的控制端。考虑到模块的复用可以减少开发难度和成本，该系统应同样借助机载 WiFi 模块回传图像数据，并且将图像显示在手机 APP 上。此外，机载摄像头应具有重量轻，体积小，耗电量低的特点。

2.2 系统总体架构

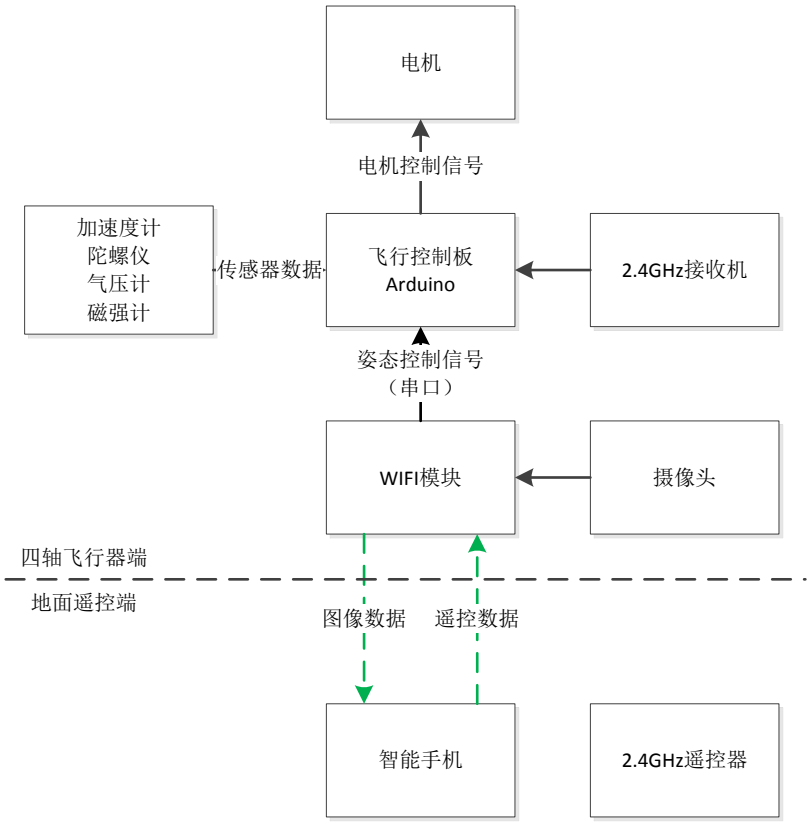

图 2.2 飞行器总体架构图

整个系统的流程如图 2.2 飞行器总体架构图所示，文字描述如下：

1. 运行在智能手机上的遥控 APP 发出遥控指令；
2. WiFi 模块接收到遥控指令后，通过串口传递给飞控板；

3. 飞控板接收到遥控指令，结合通过传感器数据解算出的姿态数据，计算得出四个电机的转速，并驱动电机作出响应；
4. 同时，摄像头将拍摄到的图像信息通过 USB 传给 WiFi 模块，WiFi 模块再通过 WiFi 转发给智能手机。

图中的 2.4GHz 接收机与 2.4GHz 遥控器是飞行器的备用控制系统，当 WiFi 控制系统发生故障时(比如手机死机，WiFi 断开连接，WiFi 模块掉电等)，控制者可以迅速启用 2.4GHz 遥控器控制飞行器，保证其平稳地降落。

2.3 四旋翼飞行器实现方案

2.3.1 组成部件

四旋翼飞行器可以分为三个部分：动力装置、支撑结构、控制电路。

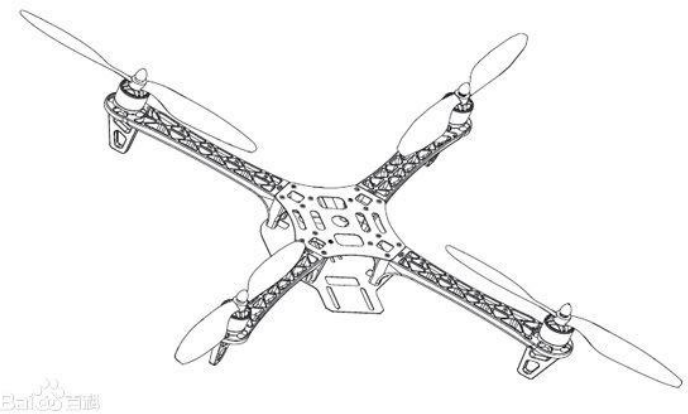

图 2.3 四旋翼飞行器示意图

动力部分由四个搭配有螺旋桨的电机组成。其中两支螺旋桨为正桨（逆时针旋转），两支为反桨（逆时针旋转），以此达到扭矩平衡。飞行器靠四支螺旋桨的旋转来提供升力。飞行器的四支螺旋桨都直接固定在交流无刷电机上，与直升机的主螺旋桨不同，两者中间并没有复杂的传动装置。

支撑结构由四个机臂和一个中央的平台构成，是整个飞行器的骨架，也为动力装置和控制电路提供放置空间。机臂采用轻便且强度高的材料，比如玻璃纤维或碳纤维。四个机臂的末端固定着装有螺旋桨的电机，另一端与中央的平

台相连。

控制电路由飞行控制板（以下简称飞控板）、遥控接收机、电子调速器（以下简称电调）和电池构成。

飞控板是一块集成了单片机和各种传感器的印刷电路板。传感器包括三轴陀螺仪、三轴加速度计、磁强计、气压计等。它们采集到的数据被单片机用来进行飞行器飞行姿态的解算。

遥控接收机接收地面遥控器传来的遥控信号，并把遥控信号传递给飞控板。现在主流的遥控信号为基于 2.4GHz 的脉冲位置调制(PPM)信号。

电调与电池直接相连，它把 11V-12V 的输入电压降为 5V，为飞控板供电。并且电调从飞控板里读取控制电机的脉宽调变信号(PWM)，转化为驱动电机的电平信号。每个电机对应一个电调。四旋翼飞行器上共有四个电调。

四旋翼飞行器的电机功率较大，输入电流较大。所以飞行器上的电池一般选用可以快速放电的动力 Li-Po 电池。该电池一般为多个 3.7V 的 Li-Po 电池串联而成，常见的有两节串联(2S)至六节串联(6S)。本项目中采用三节串联(3S)。

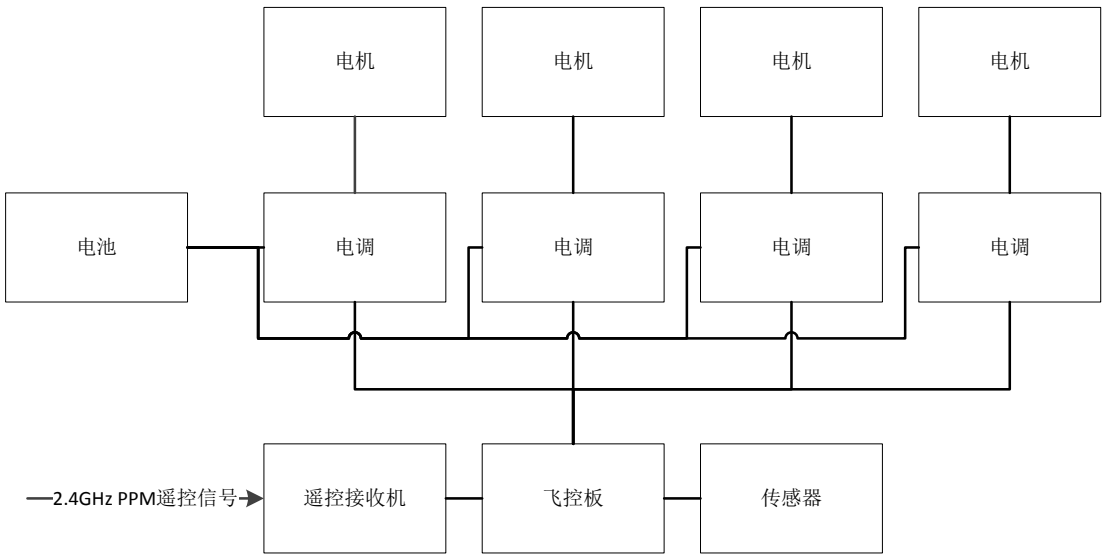

图 2.4 四旋翼飞行器各模块连接示意图

2.3.2 运动原理

四轴飞行器通过调整四个电机的转速来进行运动。四个电机的不同转速可

以为飞行器提供不同的升力和力矩，由此控制四轴飞行器的姿态。图 2.5 是四旋翼飞行器的坐标系示意图，其中 X 轴的正方向为飞行器的前进方向.

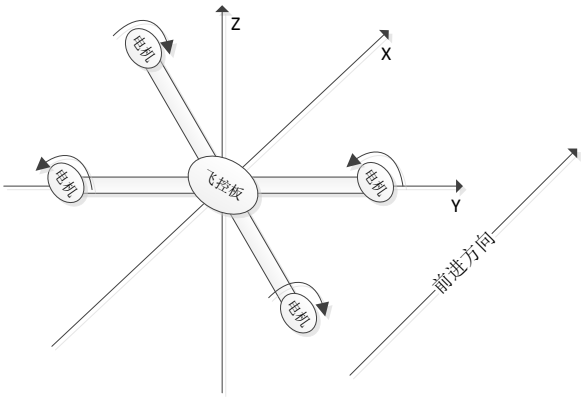

图 2.5 飞行器坐标系

其中，右前方和左下方的电机逆时针旋转，左前方和右下方的电机顺时针旋转。这样设计是为了将电机旋转所产生的力矩抵消。当四个电机的转速相等时，整个飞行器不会发生旋转。四轴飞行器有四个控制自由度，即四个电机的转速。但它在空间中有六个活动自由度，即沿着三个坐标轴作平移和旋转动作。它是一个非线性，多变量，高度耦合的欠驱动系统[5]。它有四种 运动方式，分别是升降运动，俯仰运动(pitch)，横滚运动(roll)，偏航运动(yaw)。其中，俯仰运动耦合了飞行器沿 X 轴方向的平移和沿 Y 轴方向的旋转；横滚运动耦合了飞行器沿 Y 轴方向的平移和沿 X 轴方向的旋转。下面将对四种运动方式进行详细介绍。

2.3.2.1 升降运动

当四个电机的转速均匀增加时，飞行器所获得的升力大于自身重力，飞行器获得向上的加速度，飞行器做减速下降运动或加速上升运动；当四个电机的转速均匀减小时，飞行器所获得的升力小于自身重力，飞行器获得向下的加速度，飞行器做减速上升运动或加速下降运动。

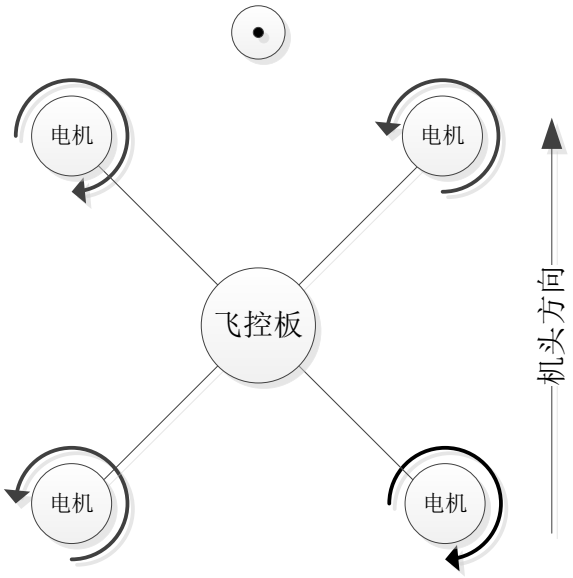

图 2.6 上升运动示意图

2.3.2.2 俯仰运动

当飞行器前方两个电机的转速下降，后方两个电机的转速上升时，飞行器后部获得的升力大于其前部获得的升力，飞行器获得一个使它向前旋转的力矩，绕 Y 轴旋转，当飞行器向前倾斜一定角度后，飞行器电机所提供的升力不再竖直向上，而是指向斜上方，这样升力就会给飞行器提供一个向前的分力，使飞行器向前运动。于此类似的，当飞行器前方两个电机转速上升，后方两个电机转速下降时，飞行器向后运动。

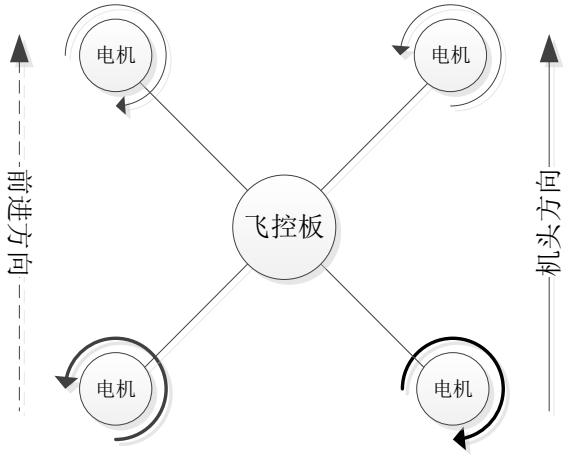

图 2.7 前倾运动示意图

2.3.2.3 横滚运动

与俯仰运动类似，当飞行器左侧的两个电机转速下降，右侧的两个电机转速上升时，飞行器沿 X 轴逆时针旋转，并向左侧飞行；当飞行器左侧的两个电机转速上升，右侧的两个电机转速下降，飞行器沿 X 轴顺时针旋转，并向右侧飞行。

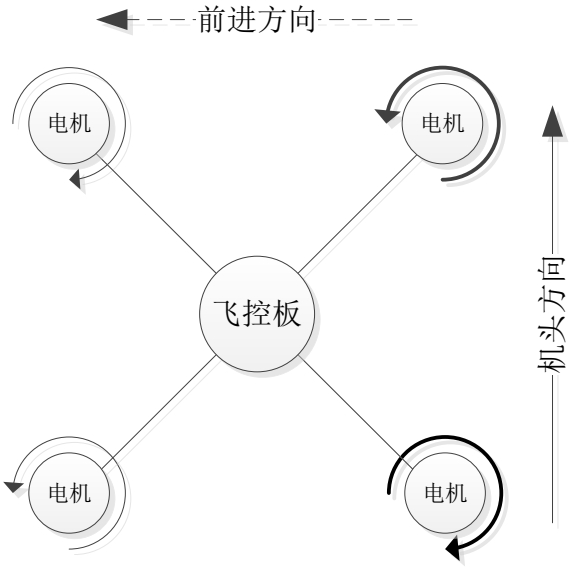

图 2.8 左横滚运动示意图

2.3.2.4 偏航运动

当飞行器的顺时针旋转的电机（左上和右下）转速增大，逆时针旋转的电

机（右上和左下）转速减小，且总升力不变时，电机产生的反扭矩不平衡，飞行器就会以自身为中心，在 XY 平面逆时针旋转；当顺时针旋转的电机转速减小，逆时针旋转的电机转速增大时，飞行器就会在 XY 平面顺时针旋转[6]。

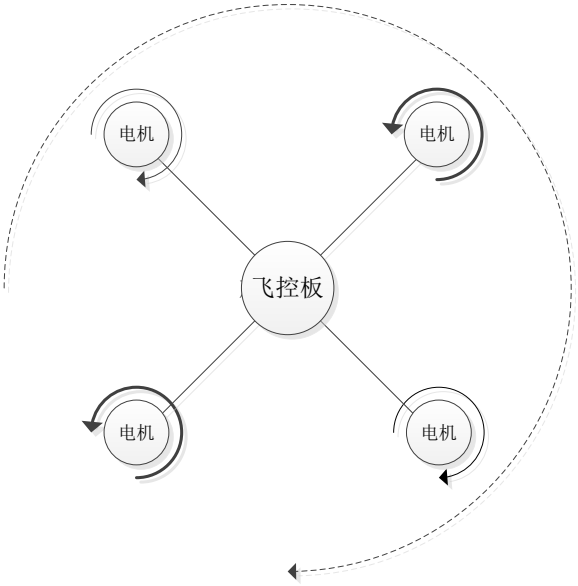

图 2.9 右偏航运动示意图

2.3.3 Multiwii 开源飞控项目

Multiwii 是由法国的无线电爱好者 Alex 发起的开源四轴飞行器控制项目。该项目使用 Arduino 板作为主处理器，传感器则可以任意搭配。该项目旨在简化电子器件的组装过程。它使用任天堂开发的 Wii 上现成的加速度计和陀螺仪。该项目支持板载的 GPIO 口输入控制信号，同样支持串口输入控制信号[7]。

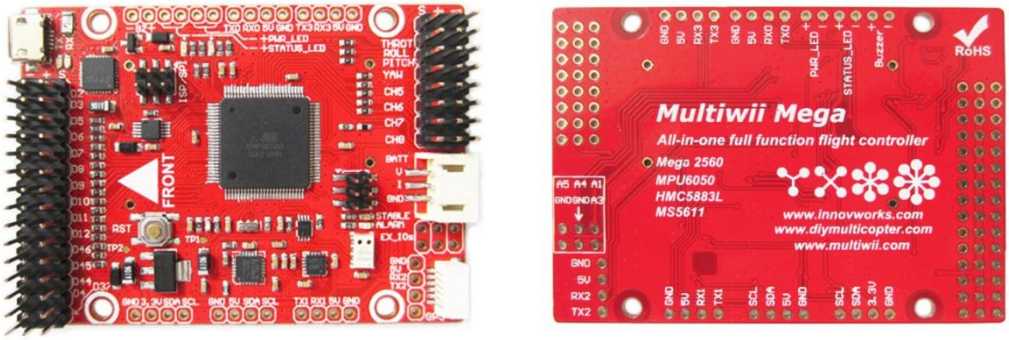

图 2.10 Multiwii 飞控板

2.4 WiFi 模块实现方案

根据 WiFi 模块的系统需求分析，并考虑到成本及可扩展性，该项目选用了 TPLINK 出产的 TL-WR703N 路由器作为 WiFi 模块。该路由器具有体积小，耗电低，重量轻等优点。项目中将该路由器刷入 Openwrt 并安装 ser2net 包，实现网口到串口的转发。

TL-WR703N 路由器自带一个 USB 口，可以与 Multiwii 飞控板的 microUSB 口相连，实现二者间的通信。

供电方面，该路由器与飞控板都是 5V 直流供电。路由器直接从飞控板的 VCC 和 GND 口取电。

图 2.11 TL-WR703N 路由器

2.5 手机端遥控 APP 实现方案

手机端 APP 采用 Android JDK 编写，适用于 Android 手机及 Android 平板。

该 APP 是在 Multiwii 论坛上的爱好者 electronicguy 的项目基础上进行开发，本项目解决了原项目中的一些 bug，并增加了多点触控，延迟时间监控，手机姿态控制等功能。

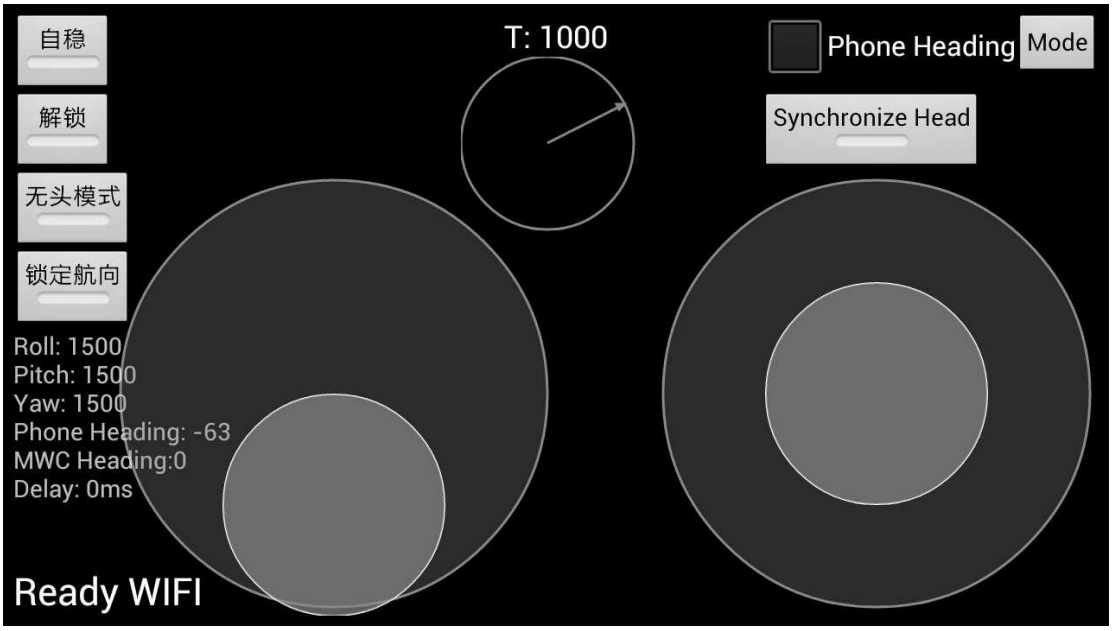

图 2.12 手机 APP 界面

2.6 图像实时回传系统实现方案

本项目使用了 iMac 一体机上的摄像头，封装双屏蔽线后可以直接连接到路由器的USB端口上。此处需要连接一个USB HUB来扩展路由器的USB口数目，让它能够同时连接摄像头和飞控板。iMac 的摄像头是免驱动的，但 Openwrt 需要安装一个mjpeg-streamr软件来读取摄像头数据并转发到默认端口号为8080的网页中，供手机端读取。

手机端方面，需要在 APP 中整合一个图像显示模块，实时显示摄像头的回传数据。

图 2.13 iMac 一体机摄像头

第3章 在项目中负责的具体工作

3.1 四旋翼飞行器系统的详细实现

3.1.1 硬件模块

3.1.1.1 电机

飞行器的升力全靠电机带动螺旋桨提供，为了获取高性能的升力，项目中选择了交流无刷电机，型号为 Sunnysky 朗宇 V2216 KV900/800。

图 3.14 朗宇交流无刷电机

3.1.1.2 螺旋桨

螺旋桨为飞行器的耗材，容易损坏。但螺旋桨的质量又关乎飞行器的升力。所以本项目选择了低价的且不失性能的仿 APC1147 螺旋桨。

图 3.15 仿 APC1147 多旋翼螺旋桨

3.1.1.3 遥控器&接收机

虽然本项目旨在搭建基于 WiFi 通信的飞行器遥控系统，但该项目的基础是搭建一个传统的基于 2.4GHzPPM 调制的四旋翼飞行器。所以购买一套 2.4GHz 的遥控设备在该项目的开发过程中仍有其必要性。其次，2.4GHz 遥控器还可以作为该项目中的备用遥控系统，以提高飞行器遥控的可靠性。

遥控器至少有四个通道，来对应飞行器的升降、偏航、横滚、俯仰运动。但是，遥控器最好有六个及以上的通道，多余的通道用来设置飞行模式。

本项目中采用了天地飞 6 通道遥控器。

图 3.16 天地飞 6 通道遥控器

3.1.1.4 电子调速器

电池的输出电压为 11V-12V 之间，电子调速器连接电池、电机与飞控板。它一方面负责把电池电压降为 5V 给飞控板供电，一方面把电池的直流电逆变成交流电供电机使用，最后还把飞控板输入的 PWM(脉宽调制)信号转换为电流大小，调节电机的转速。

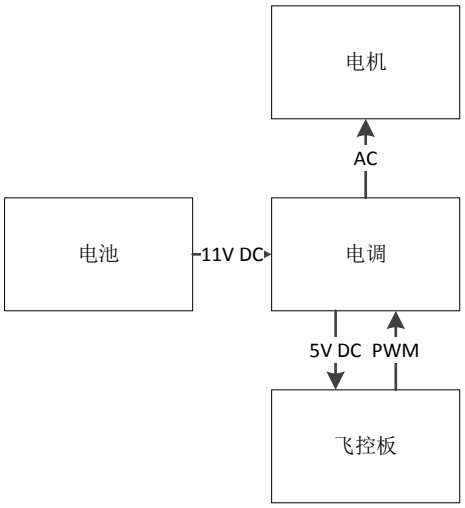

图 3.17 电调连接示意图

3.1.1.5 充电器&电池

飞行器的电池应选用高放电倍率的动力电池，电池的容量越大飞行器的续航能力越强。但大容量的电池本身质量较大，过重的电池会增大飞行器负载导致续航时间下降。所以选择合适的电池容量也是一个重要的课题。

经过在航模论坛的调研后，本项目选用了格式5C2200mAh，3S 充电锂电池。

图 3.18 格式锂电池

3.1.1.6 飞控板

本项目选择了 Multiwii 的高配红板飞控板。该飞控板的硬件参数如表格 3.1 Multiwii 飞控板的硬件参数所示。

表格 3.1 Multiwii 飞控板的硬件参数

部件	处理器	陀螺仪	加速度计	磁强计	气压计
型号	ATmega2560	ITG3200	BMA180	HMC5883L	BMP180

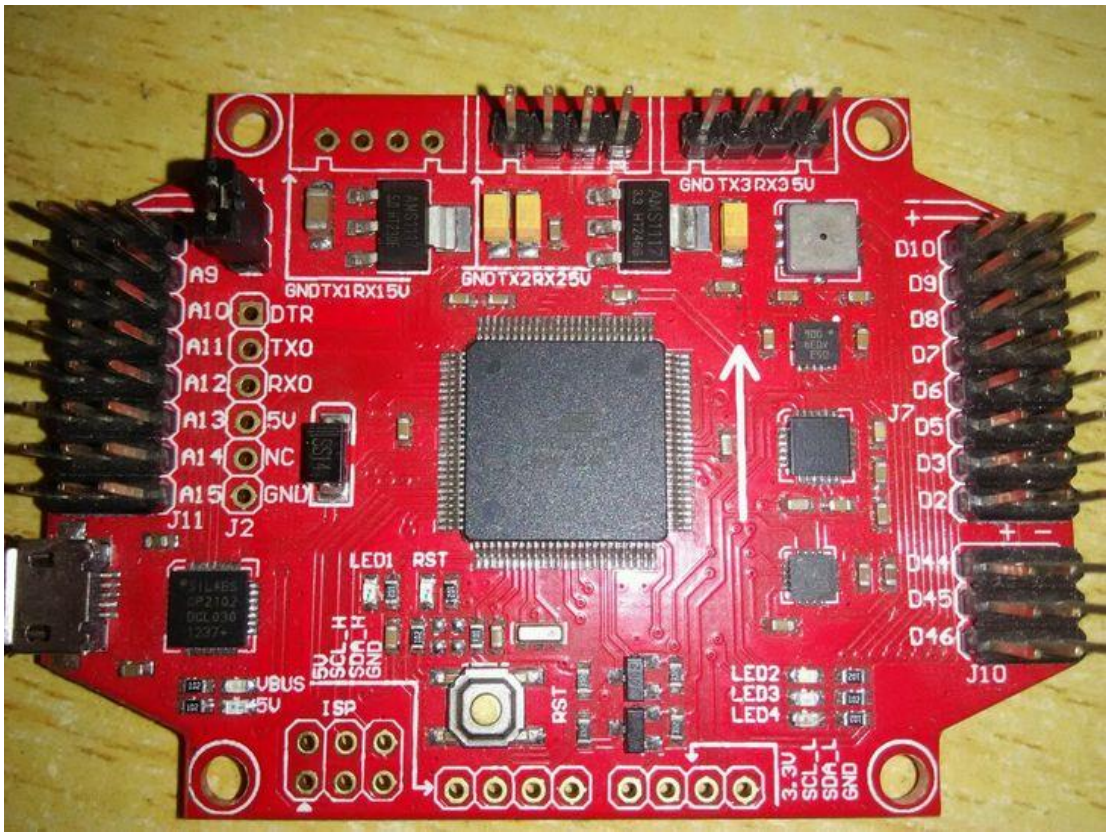

图 3.19 Multiwii 飞控板

3.1.1.7 机架

机架作为整个飞行器的支撑部分，需要具有质量小，强度大的特点。本项目采用了大疆的 450mm 玻璃纤维机架。

3.1.1.8 其他零部件

低压报警器：当电池电压过低时，该报警器发出声响，提醒用户应及时降落飞机。

飞控连接线：即杜邦线，连接飞控及其他模块。

3.1.2 软件算法

3.1.2.1 Multiwii 开源项目

Multiwii 是基于 Arduino 开发的开源四旋翼飞行器飞控项目。在 Multiwii 官网下载好最新的源码，用 Arduino IDE 打开，进行相关配置后，将飞控板与电脑连接，将固件刷入飞控板内。

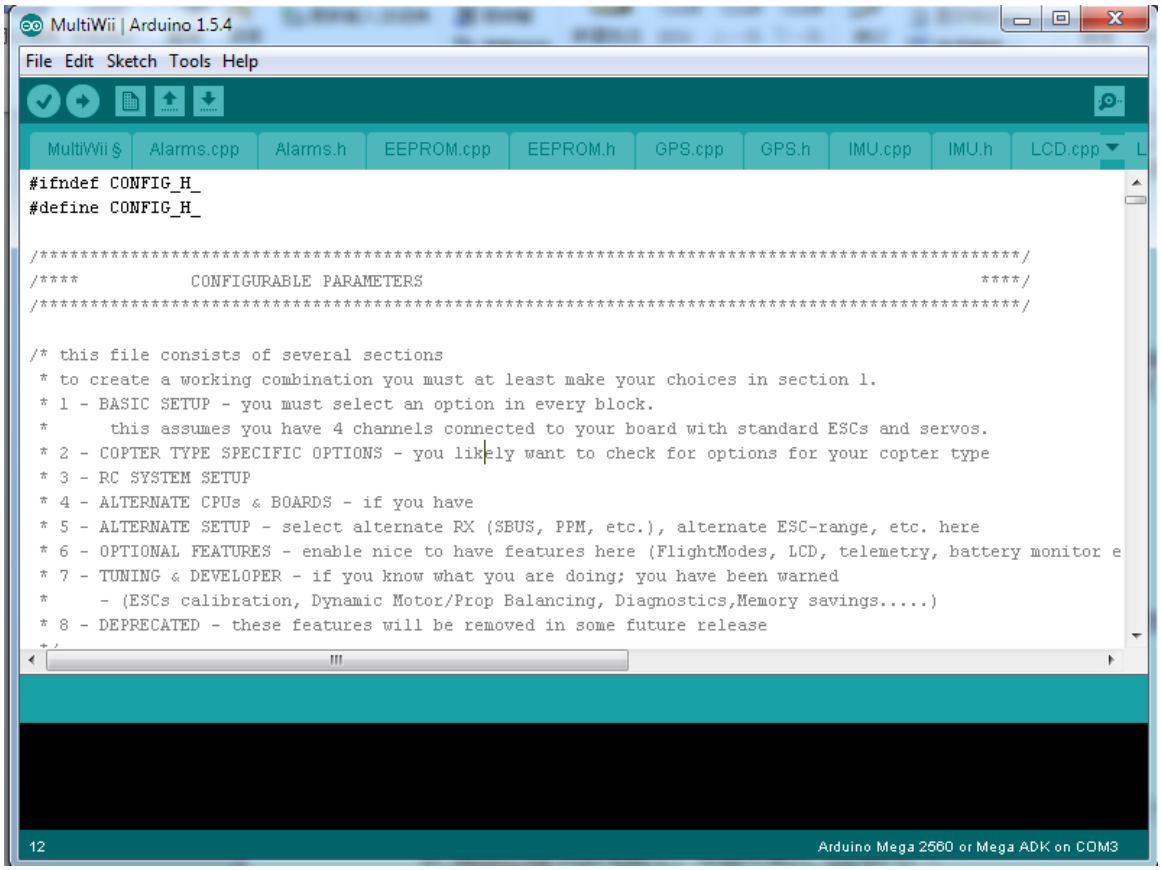

图 3.20 Multiwii 固件的编译环境 Arduino IDE

3.1.2.2 PID 控制算法

由于四轴飞行器具有不稳定、非线性、强耦合等特性，所以需要建立完整的数学模型进行分析。而姿态控制是四轴飞行器控制系统的核心[8]。所以需要在飞控板算法内构建一套自动控制算法，实时修正姿态偏差。Multiwii 中采用的是控制系统中比较流行的比例积分微分控制(PID)算法[9]。由此还衍生出了用于飞行器负载变化时进行控制的模糊 PID 控制等算法[10]。下面对 PID 算法进行简单介绍。

比例控制

比例控制器的输出与偏差成比例：

$$u(t) = K_c e(t)$$

其中，u(t)为控制器的输出，e(t)为设定值和测量值之差，Kc 为控制器的增益，通常无量纲。

其阶跃响应曲线如图 3.21 所示：

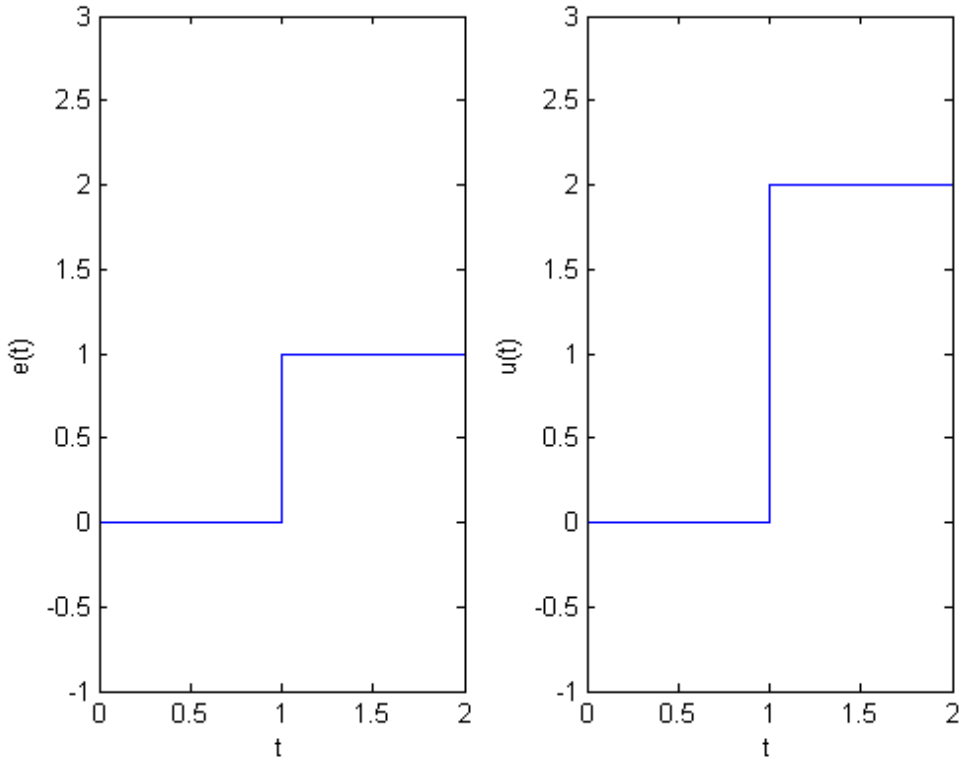

图 3.21 比例控制阶跃响应曲线

图 3.21 中，左图为控制器的输入，即 e(t)，它在 t=1 时发生由 0 到 1 的突变。右图为控制器的输出，即 u(t)，它在输入发生突变时也立刻发生等比例的变化，在本例中，Kc 的值为 2。

纯比例控制器有一个缺点就是当设定值改变后总是存在一定的余差。因此在实际使用中常采用带有积分作用的控制器，即下一节介绍的比例积分控制

比例积分控制

积分作用的输出是误差相对于时间的积分：

$$u(t) = \frac{1}{T_i}\int_0^t e(t)dt$$

其中，Ti 是积分时间。积分作用的优点是它能够消除余差。但积分作用比较慢，通常不单独使用，而是和比例作用一起使用，比例积分控制的算式为：

$$u(t) = K_c(e(t) + \frac{1}{T_i}\int_0^t e(t)dt)$$

其阶跃响应曲线为：

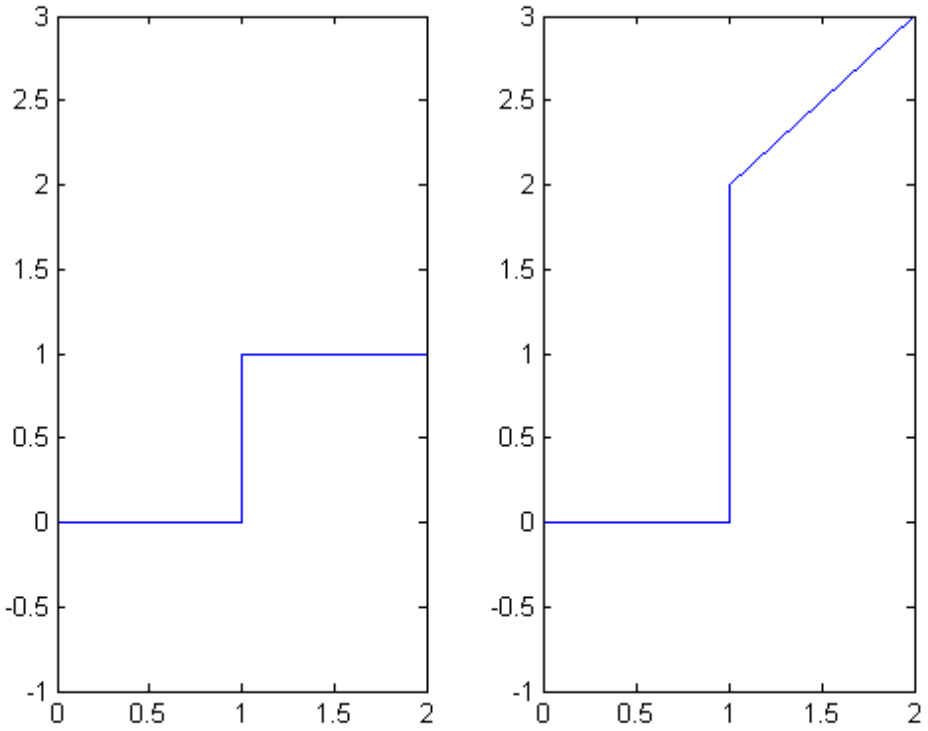

图 3.22 比例积分控制阶跃响应曲线

控制器输入在 1s 发生突变，并在 1s 之后保持不变。控制器输出在 1s 时等比例跳变至 2。由于有积分作用的存在，控制器输出持续增加。

比例积分微分控制

微分控制作用是通过误差的变化率来预报误差信号的未来变化趋势。理想的微分控制作用是：

$$u(t) = T_d \frac{de(t)}{dt}$$

其中，Td 是微分时间。当误差是常数时，u(t)=0，微分控制器没有输出。因此微分作用不单独使用，总是与比例或比例积分作用同时使用。

一个理想的 PID 控制器为：

$$u(t) = K_c(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d \frac{de(t)}{dt})$$

Multiwii 就是通过实时计算从遥控器接收到的遥控数据与当前飞行器姿态数据的差值，即 e(t)，通过 PID 算法算出 u(t)，然后将 u(t)作用到电机上。

需要指出的是，Multiwii 需要将姿态数据投影到三个正交的坐标轴上 x,y,z。然后在三个轴上分别计算出 roll,pitch,yaw 的旋转角度，然后进行 PID 控制。

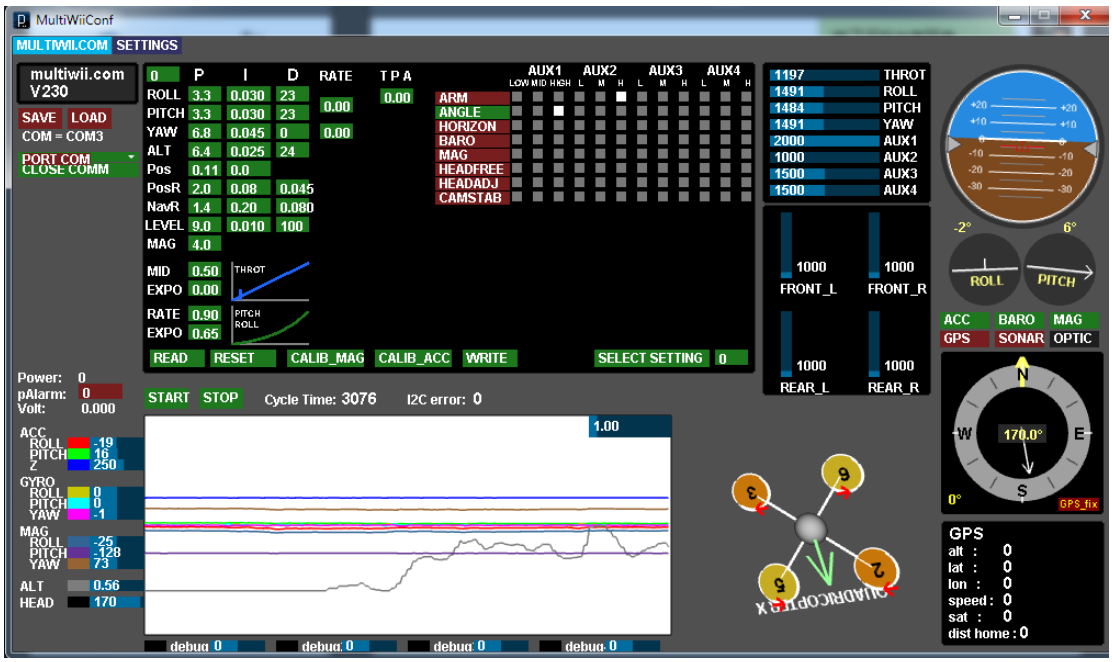

图 3.23 Multiwii 的调参界面，其中可以设置 PID 的参数

3.2 WiFi-串行信号转换系统实现方案

3.2.1 TL-WR703N 路由器

TL-WR703N 路由器由于其自带一个 USB 口，且能够刷入 Openwrt，一个完全可读写的嵌入式 Linux 系统，一经问世就收到了电子爱好者的追捧。本实验采用了该款路由器，并刷入 Openwrt 固件，让它成为完全可定制的 WiFi 模块。网上有很多刷写固件的教程，本文不再赘述。

3.2.2 Openwrt

Openwrt 是一款嵌入式的 Linux 发行版。OpenWrt 提供了一个完全可写的文件系统，从应用程序供应商下载软件包，并允许用户自定义设备，以适应任何应用程序。

刷入 Openwrt 后，电脑需要先连接路由器的 WiFi，登录网页如果(默认地址为 192.168.1.1)，新建 root 密码。之后就可以用 root 账号 ssh 到路由器进行远程操作。上传文件则需借助 winSCP 软件(Linux 下为 scp)。

图 3.24 Openwrt terminal

3.2.3 Ser2net

Ser2net 是 Openwrt 下的一个软件包。它实现了串口和网口的数据通信，能够把 TCP 协议的网口数据以指定的波特率转发到某一串口上，也可以把串口数据通过 TCP 协议从网口发送出去。

如果路由器已经连接到互联网，则在 Terminal 下输入:

Opkg update

Opkg install ser2net

如果没有连接到互联网，则需要手动在 openwrt 的软件源 http://downloads.openwrt.org/ 下载，并上传到路由器中，然后运行

Opkg install xxxxx.ipk (软件包名)

该安装包的配置文件为 etc/ser2net.conf。

打开该文件，进行配置。

```
3001:raw:600:/dev/ttyUSB0:115200 NONE 1STOPBIT 8DATABITS -XONXOFF LOCAL -RTSCTS
```

图 3.25 etc/ser2net.conf 文件的配置信息

各项参数的含义如下：

3001 – 监听的端口号；

Raw – 原始数据流；

600 – 超时时间，单位秒。如果 600 秒内没有连接，则关闭该端口；

/dev/ttyUSB0 – 转发到的串行设备名称；

115200 – 串行设备的波特率

NONE – 设置等价位(parity)

1STOPBIT – 一个停止位

8DATABITS – 八个数据位

-XONXOFF – 关闭 XON/XOFF 支持

LOCAL – 不监视调制解调器线(modem line)

-RTSCTS – 关闭硬件的流控制

3.3 手机端遥控系统实现方案

3.3.1 程序流程图

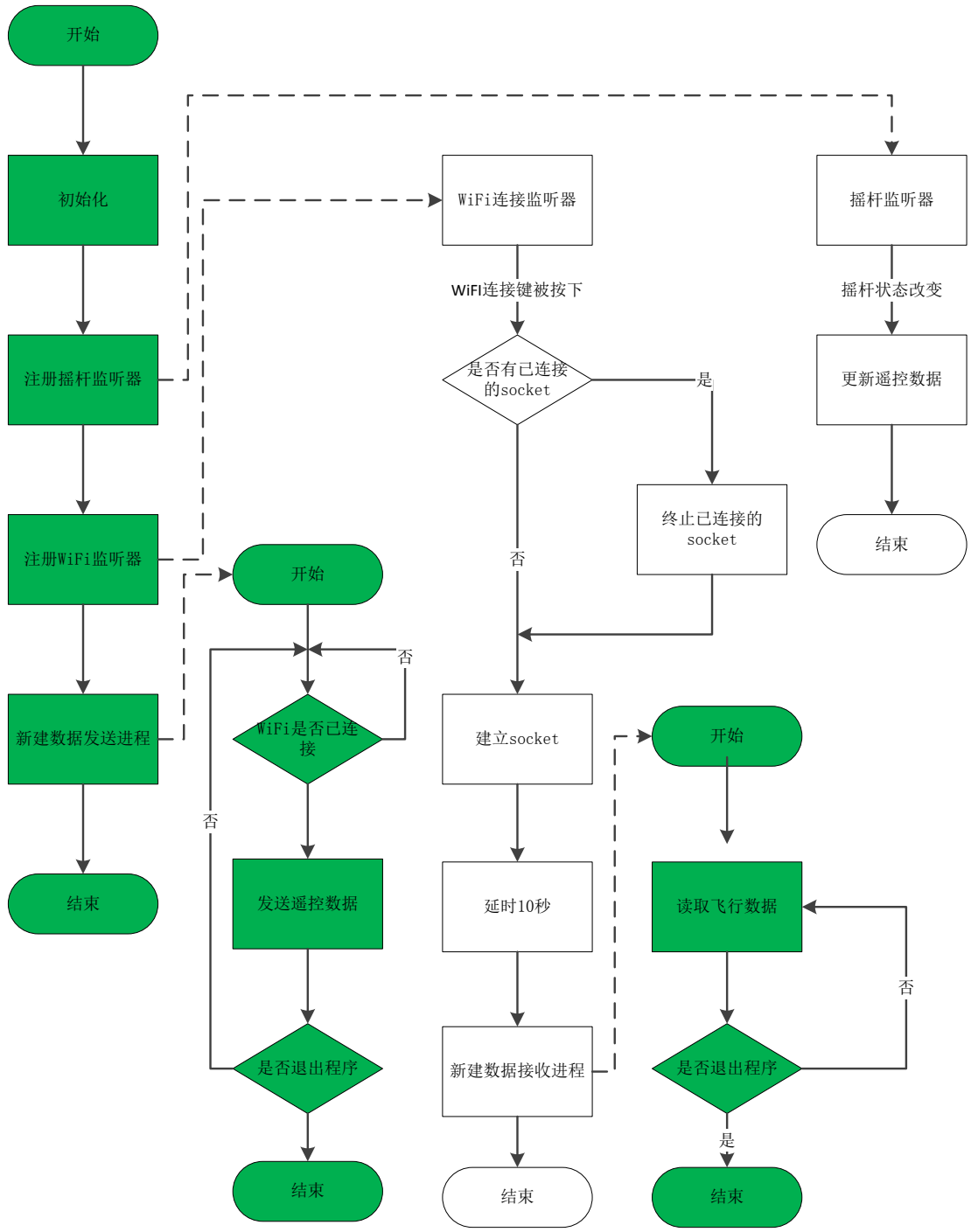

图 3.26 程序流程图。其中绿色框图代表线程，白色框图代表监听器

如图 3.26 程序流程图。其中绿色框图代表线程，白色框图代表监听器所示，图中最左一列为主线程，该线程在完成初始化界面，注册监听器和新建数据发送线程后就随之销毁。

数据发送线程会实时监测 WiFi 是否已连接，如果已连接，则读取当前的遥控数据，通过 socket 发送出去。

当 WiFi 连接按钮被按下时，WiFi 监听器先尝试终止以前连接的 socket，新建 socket 连接并等待 10 秒。此处等待的作用是等待 Multiwii 初始化完成，如进行水平校准，加速度计校准等工作。等待 10 秒后新建数据接收进程，用来实时读取飞行器回传的飞行数据。

当用户触碰摇杆改变遥控指令时，摇杆监听器会更新位于内存中的遥控数据，供数据发送进程进行发送。

3.3.2 MVC 设计原则

MVC 即为 Model-View-Controller 的缩写。MVC 模型把软件划分成了三个核心模块，视图(View)负责把 Model 展示给用户，控制器(Controller)负责让用户改变 Model，模型(Model)负责储存数据[11]。本 APP 的开发采用了 MVC 设计原则，将各个模块拆分，以减少各个模块间的耦合，降低维护成本。

3.3.3 Model-遥控数据协议

Multiwii 支持通过串口读取遥控信息控制飞行器，根据 Multiwii 的作者 Alex 在 Multiwii 官网公布的信息，其串口协议描述如下：

Multiwii 串口协议(MSP)的一般格式为<先行符><方向><大小><命令><数据><校验码>

其中先行符是两个 ASCII 字符”$M”用来标记数据的开始；

<方向>中，”<”表示数据是从控制端发向飞行器，”>”表示数据从飞行器发向控制端。

<大小>的值表示<数据>段的字节数。如果当前数据包不包括<数据段>，比如是一个发向飞行器请求飞行数据的数据包，那么大小就置为 0。

<命令>是一个用来区分不同指令的 32 位整数。

<数据>中包括了该种命令传递的数据，可为空。数据的格式因<命令>而异。

校验码是对<大小><命令>和<数据>段中的每个子节进行异或操作的结果。

详细的协议文件在 Muliwii 的 Wiki 上可见。
http://www.multiwii.com/wiki/index.php?title=Multiwii_Serial_Protocol

```
public List<Byte> requestMSP(int msp, Character[] payload) {
    if (msp < 0) {
        return null;
    }
    List<Byte> bf = new LinkedList<Byte>();
    for (byte c : MSP_HEADER.getBytes()) {
        bf.add(c);
    }

    byte checksum = 0;
    byte pl_size = (byte) ((payload != null ? (int) (payload.length) : 0) & 0xFF);
    bf.add(pl_size);
    checksum ^= (pl_size & 0xFF);

    bf.add((byte) (msp & 0xFF));
    checksum ^= (msp & 0xFF);

    if (payload != null) {
        for (char c : payload) {
            bf.add((byte) (c & 0xFF));
            checksum ^= (c & 0xFF);
        }
    }
    bf.add(checksum);
    return (bf);
```

图 3.27 发送带有数据的 request 代码片段

3.3.4 View-用户视图

View 负责给用户提供可视化的界面。在 Android 开发中，view 的代码基本放在 view 包中，其布局通过 layout/*.xml 文件控制。

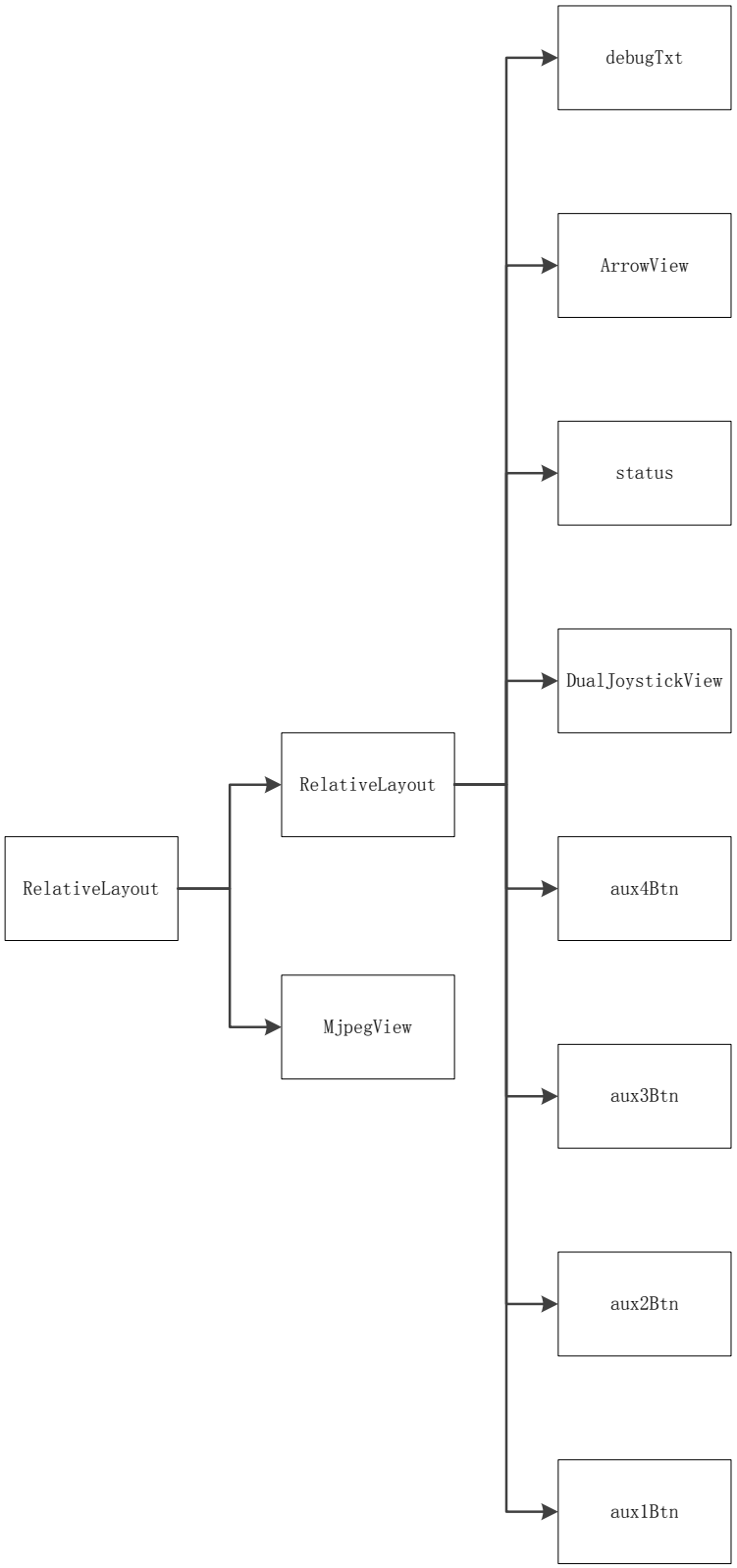

图 3.28 view 架构图

其中，各个 view 之间的布局采用 relativeLayout，MjpegView 为图像回传显示的视图，aux1Btn~aux4Btn 为遥控器的四个辅助通道，用来设置飞行模式，如无头模式，自稳模式，解锁等。Status 和 debugtxt 为显示的调试信息。ArrowView 显示机头的朝向。DualJoystickView 显示摇杆。

3.3.5 Controller

3.3.5.1 WiFi 连接模块

该模块负责与机载的 WiFi 模块-TL-WR703N 路由器建立 TCP 连接。建立连接时需要指定目标地址的 IP 及端口号，来建立 socket。并且把 socket 传递给数据发送线程和数据接收线程。

```
@Override
public boolean Connect(String ip, int port) {
    Enable();
    address = ip + ":" + port;
    setState(STATE_CONNECTING);
    try {
        mySocket = new Socket(ip, port);
        mySocket.setKeepAlive(true);
        inStream = mySocket.getInputStream();
        outStream = mySocket.getOutputStream();
        setState(STATE_CONNECTED);
        //Connected = true;
    } catch (Exception e) {
        e.printStackTrace();
        connectionLost();
        return false;
    }
    //Log.d("connect", "connected!!!!!");
    return true;
}
```

图 3.29 WiFi 连接代码片段

3.3.5.2 数据发送模块

当 WiFi 模块建立后，数据发送模块负责将内存中的遥控数据发送给飞行器。同时也向飞行器请求当前机头的朝向，供机头同步模块同步机头与手机的方向，也用于计算数据发送的延迟。

此外，数据发送模块也对 Multiwii 的数据协议进行了封装，把复杂的数据格式装载封装在函数中。这样上层只要调用函数即可完成数据的发送。

3.3.5.3 数据接收模块

数据接收模块负责接收飞行器回传的飞行参数。包括当前机头的朝向信息，以及飞行器回传的遥控命令的 ACK 信息。

根据 3.3.3 中描述的 Multiwii 数据协议，每个数据包由<先行符><方向><大小><命令><数据><校验码>六个部分组成，其中，<数据>部分的大小不确定，跟<命令>有关，由<大小>的值确定，其他五个部分的大小确定。项目中采用了有限状态机的方法来读取数据。每成功读取一个字段的数据便作一次状态转移，如果读取到的数据不合法，则返回到起始状态。

```
private void ReadFrame() {
    DataFlow--;
    while (communication.dataAvailable()) {
        //It's a finite state machine ..
        try {
            c = (communication.Read());
        } catch (Exception e) {
            c_state = IDLE;
            Log.e("MultiwiiProtocol", "Read  = null");
            break;
        }
        if (c_state == IDLE) {
            c_state = (c == '$') ? HEADER_START : IDLE;
        } else if (c_state == HEADER_START) {
            c_state = (c == 'M') ? HEADER_M : IDLE;
        } else if (c_state == HEADER_M) {
            if (c == '>') {
                c_state = HEADER_ARROW;
            } else if (c == '!') {
                c_state = HEADER_ERR;
            } else {
                c_state = IDLE;
            }
        } else if (c_state == HEADER_ARROW || c_state == HEADER_ERR) {
            /* is this an error message? */
            err_rcvd = (c_state == HEADER_ERR); /*
                                                 * now we are expecting the
```

图 3.30 数据读取模块代码片段

3.3.5.4 多点触控模块

如图 2.12 所示，摇杆界面由两个摇杆组成，左侧摇杆控制油门与偏航，右侧摇杆控制俯仰与横滚。为了能让用户同时更改这两个摇杆的状态，必须加入多点触控的支持。

在 Android 开发中，触控事件被封装为 MotionEvent 类，用户的每个触碰事件都会生成一个 MotionEvent 对象，传递给用户触碰的 view，由 view 中的函数 onTouchEvent 负责处理。

在项目的实现中，摇杆的触控事件统一由 DualJoystickView 类处理，该类再把触控事件分发给左侧摇杆和右侧摇杆，相关代码如图 3.31 所示

```
@Override
public boolean dispatchTouchEvent(MotionEvent ev){
    boolean l = stickL.dispatchTouchEvent(ev);
    boolean r = stickR.dispatchTouchEvent(ev);
    return true;
}
```

图 3.31 触控事件分发代码

在单点触控中，用户按下手指，滑动手指，抬离手指分别对应 MotionEvent 中的 ACTION_DOWN，ACTION_MOVE，ACTION_UP 类型。

在多点触控中，当已有一个点按下时，再按下一个点则会触发 ACTION_POINTER_DOWN 事件，滑动会触发 ACTION_MOVE 事件，抬起会触发 ACTION_POINTER_UP 事件[12]。

在程序中则用了一个 switch 来判断不同的事件，然后判断触发位置是否在该 view 的范围内，最后进行相关的判定。

3.3.5.5 姿态解算模块

该项目实现了通过手机的姿态(重力传感器)来控制飞行器的姿态的功能，让用户通过手机控制飞行器变得更加轻松。

在 Android 中，通过注册一个 ROTATION_VECTOR_SENSOR，并注册相应的 onSensorChangedListener，我们可以获得描述当前手机姿态的四元数。然后经过计算把四元数转换成旋转矩阵，再转换成描述手机在 Z 轴，Y 轴，X 轴旋转角度的方向矩阵，就完成了对手机姿态的解算[13]。

最后把手机的姿态映射到 pitch 和 roll 的摇杆偏移量，我们就实现了用手机姿态操控摇杆，进而操控飞机姿态的功能。

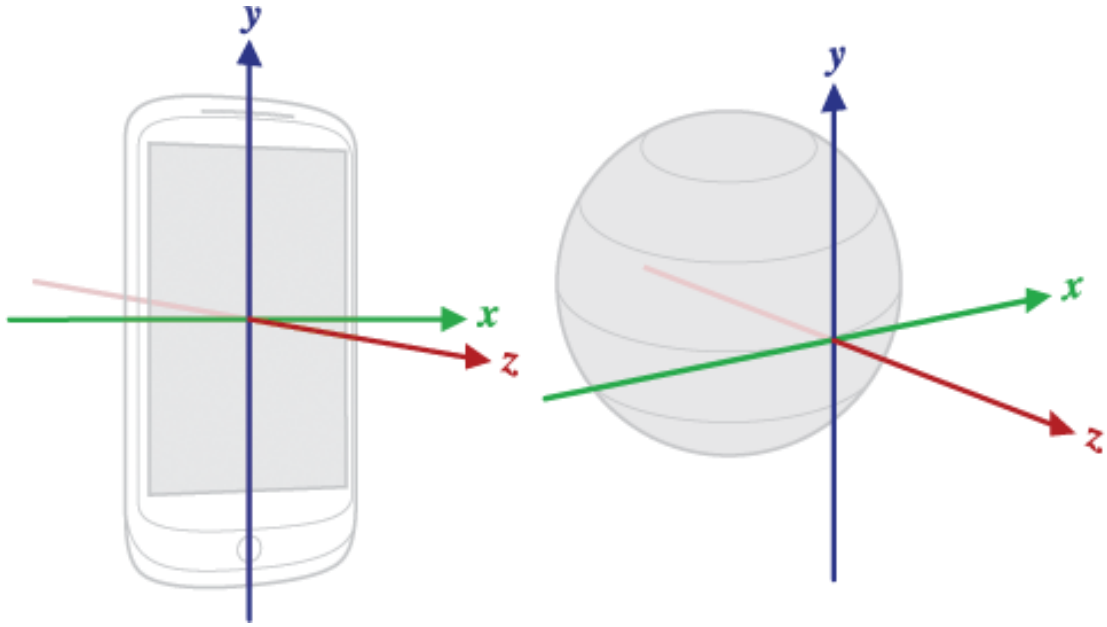

图 3.32 Android 定义的手机坐标系与地球坐标系。左图，手机坐标系中，y 轴正方向指向手机的长边正上方，x 轴正方向指向手机短边的右侧，z 轴指向手机屏幕正上方，方向朝外；右图，地球坐标系中，y 轴正方向指向正北，x 轴正方向指向正东，z 轴正方向指向天空

3.3.5.6 机头同步模块

3.3.5.5 节中的姿态解算模块能够将手机在 X 轴和 Y 轴的旋转角度映射到 pitch 和 roll 的摇杆偏移量。但我们却不能简单地把手机相对于 Z 轴的旋转角度映射到 yaw 上，而是需要实时计算当前飞行器机头与手机机头的偏差，然后通过 PID 算法来进行控制。从而实现飞行器机头与手机机头的朝向同步。

关于 PID 的详细介绍参见 3.1.2.2 一节，在机头同步模块中又增加了如下的特性。

增量算式

模拟 PID 控制算式为

$$u(t) = K_c(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d\frac{de(t)}{dt})$$

其中，Kc，Ti，Td 分别为模拟控制器的比例系数，积分时间和微分时间。e(t)为 t 时刻的设定值和测量值之间的差值，u(t)是模拟控制器在 t 时刻的输出。因为计算机处理的是数字信号，需要将上式进行离散化，用累加代替积分，用差值代替微分，离散化后的算式为：

$$\mathrm{u(k)} = \mathrm{K_c}(e(k) + \frac{T_s}{T_i}\sum_{i=0}^{k} e(i) + \frac{T_d}{T_s}(e(k) - e(k-1))$$

在实际运算中，上式不仅需要计算当前偏差 e(k)，还要计算以前所有偏差的和，这种算式计算量大，且需要较大的存储空间，在实际应用中，我们往往使用增量算式

$$\Delta\mathrm{u(k)} = \mathrm{u(k)} - \mathrm{u(k-1)}$$
$$= \mathrm{K_c}\big(\big(e(k) - e(k-1)\big) + \frac{T_s}{T_i}e(k) + \frac{T_d}{T_s}(e(k) - 2e(k-1) + e(k-2))\big)$$

而$\mathrm{u(k)} = \mathrm{u(k-1)} + \Delta\mathrm{u(k)}$。其中，Ts 为采样周期。

在实际测试中，发现在控制机头转向的任务中，不需要微分作用，只要比例积分作用就可以获得比较好的控制效果，所以项目中采用了 PI 控制。

设置控制器输出阈值

由比例积分的算式可知，如果 e(t)足够大，或者 e(t)存在的时间足够长，控制器输出 u(t)没有上限，可以到无穷大。但在控制系统中，控制器的输出往往是存在上下限的（阈值）。如控制飞行器偏航的 yaw 通道，它的输出上下限为 1500±500(1500 为中立点)。实际控制中，为了防止飞行器过快地偏航导致系统不稳定，我们把 yaw 通道的输出阈值设置为 1500±70 之间，如果控制器的原始输出超过了阈值，则取阈值的边界值作为控制器的最终输出。

防止积分饱和

对于一个有积分作用的控制器，只要偏差 e(t)存在，控制器的积分作用就会不断对偏差进行累计并增大控制器输出，为了让控制器的输出限定在阈值范围内，在控制器的输出达到阈值时，要关闭积分作用，避免控制器的输出过大，防止积分饱和。

3.4 图像实时回传系统实现方案

3.4.1 机载摄像头的选取

机载摄像头选取了 iMac 一体机的拆机摄像头。它的重量轻，分辨率高，免驱动且支持 MJPG-streamer 软件。项目中把摄像头固定在机头的前进方向，然后用 USB 线连接到路由器上。

3.4.2 MJPG-streamer

MJPG-streamer 是一款 Linux 下的开源软件。它可以把从 Linux-UVC 兼容的视频头内读取到的 JPG 图片整合成 M-JPEG 流传输到网页上。

安装配置过程也比较简单，首先 ssh 到路由器上安装必要的程序包

Opkg install kmod-video-uvc mjpg-streamer

Opkg 包管理系统会自动解决包依赖问题。安装结束后，将摄像头通过 USB 连接路由器，此时如果程序正常安装，我们会看到/dev/video0 设备。

开机自启动 MJPG-streamer

/etc/init.d/mjpg-streamer enable

启动 MJPG-streamer

/etc/init.d/mjpg-streamer start

之后就可以在路由器的 8080 端口看到视频流的输出[14]。

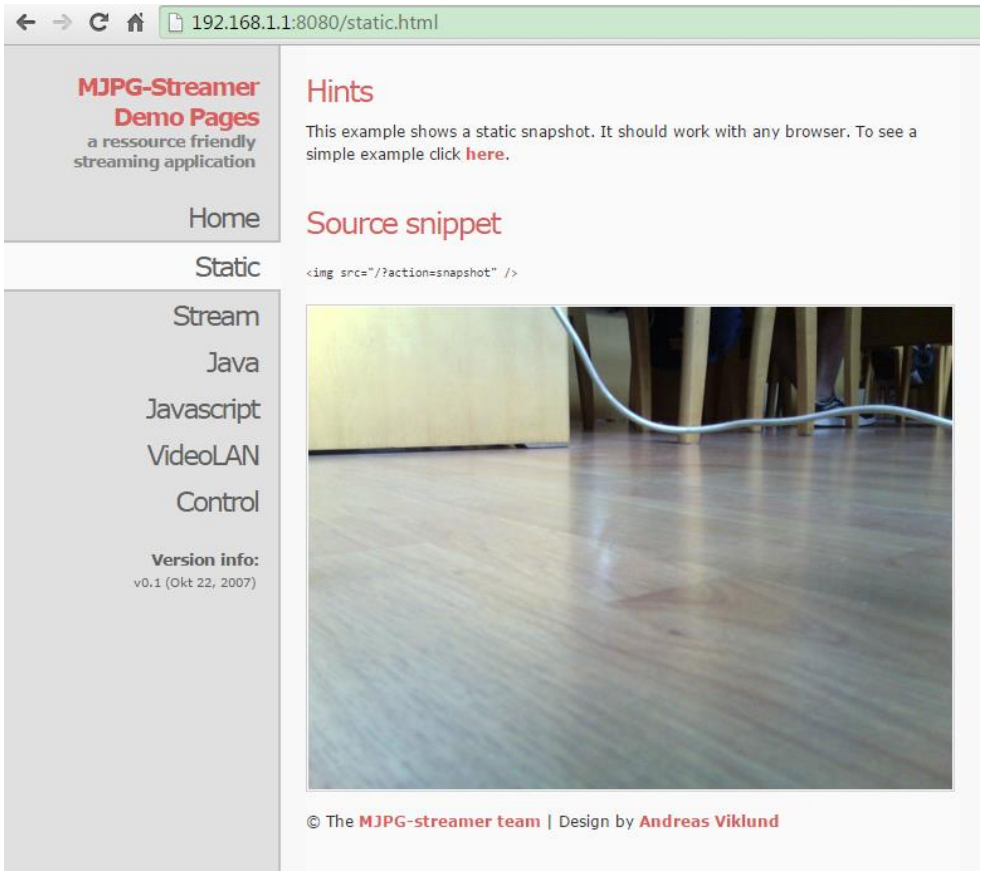

图 3.33 视频流输出

第4章 项目成果

4.1 遥控系统测评

经过试飞测试，WiFi 遥控系统能够较好地完成飞行器的控制任务。在空旷且关闭图像回传系统的情况下，最大遥控距离可达 20m 左右。数据包延迟在 10ms 左右，反应灵敏。

表格 4.2 遥控性能测评(不打开图像回传功能)

距离	2m	5m	10m	20m
延迟	7ms	7ms	8ms-13ms	10ms-113ms
丢包率	0%	0%	2%-4%	0%-43%

表格 4.3 遥控性能测评(打开图像回传功能)

距离	2m	5m	10m	20m
延迟	17ms-30ms	15ms-40ms	31ms-58ms	>100m
丢包率	0%	3%-8%	30%-58%	40%-79%

由文献[15]的研究可知，WiFi 的传输距离与接收端接收到的信号的信噪比有关，如果信号过弱，噪声过强，信噪比低于接收端解调的阈值，则接收端无法成功解调数据包，导致数据包丢失。

影响信噪比的因素有很多，发送端与接收端的距离增大，或者发送端与接收端之间有障碍物，都会降低接收端收到的信号强度，从而降低了信噪比；所处环境的干扰较大，会增加接收端的噪声强度，从而降低信噪比。

所以，用 WiFi 控制飞行器飞行时，本文建议用户一定要在远离干扰源(如发射塔)的空旷室外进行，且飞行距离不可以过远，在 20m 范围内为宜。

由表格 4.2 遥控性能测评(不打开图像回传功能)所示，随着遥控端与飞行器距离的增加，延迟和丢包率都在上升，且数据发送的不稳定性也急剧上升，在 20m 时，有时完全没有丢包，而有时丢包率达到了 43%。由表格 4.3 所示，当开启图像回传功能时，遥控信号的延迟和丢包率明显增加。遥控稳定性急剧下

降。这是因为图像数据占据了 WiFi 的带宽，遥控数据的带宽被压缩所致。

图 4.34 飞行实拍

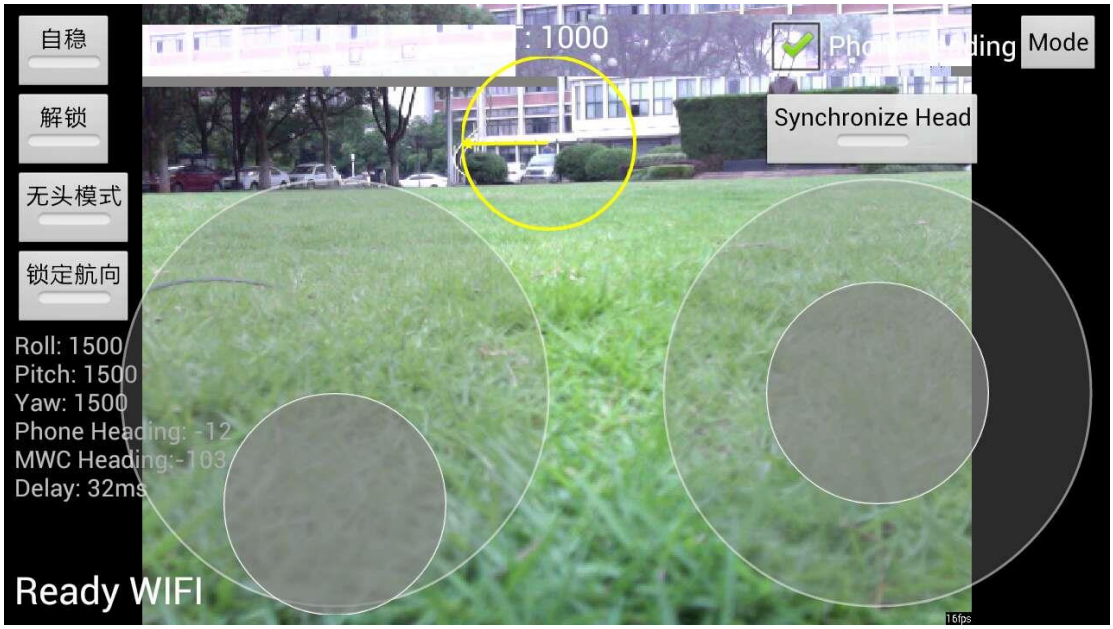

图 4.35 手机 APP 截图

实测发现，当电机开始转动后，机载罗盘工作不正常。罗盘的指向会顺时

针偏移 40 度左右，指向东北方向。因为机头同步模块会利用机载罗盘和手机内置罗盘进行方向比对，由它们的差值经由 PID 计算成为 yaw 通道的输出，所以该偏差会导致同步模块无法完成同步任务。

因为摄像头和路由器都直接从飞控板上取电，初步怀疑是罗盘传感器供电电压不足，导致罗盘工作失灵。此处仍需要改进，尝试绕过飞控板直接从电调取电。

4.2 图像回传系统测评

实测发现，图像回传系统能实时地传回飞行画面，完成了预先的任务要求。

图 4.36 航拍视频截图

但是，当开启图像回传模块时，WiFi 的大部分带宽会被图像数据占用，遥控数据的延迟变得极大，一般大于了 100ms。这样的延迟对操控飞行器来说是非常危险的，且容易导致飞行器飞行不灵敏。可以考虑将飞行视频存储在机载的内存卡上，等飞机降落后再从中读取视频。或者进一步压缩回传的图像数据，

这些都是可以改进的地方。

参考文献

[1] LEISHMAN J.The breguet-richet quadrotor helicopter of 1907

[2] 徐华中，余飞，何家俊；卡尔曼滤波在四轴飞行器导航中的应用；2012 年；武汉理工大学学报（信息与管理工程版）；第 34 卷第 3 期。

[3] Hyon Lim, Jaemann Park, Daewon Lee, and H.J. Kim. Build Your Own Quadrotor. IEEE ROBOTICS & AUTOMATION MAGAZINE, SEPTEMBER 2012.

[4] 李旭升；模型遥控 2.4GHz 系统的设计与实现：博士学位论文，电子科技大学，2010 年。

[5] Mian A A, Daobo W. Nonlinear Flight Control Strategy for an Underactuated Quadrotor Aerial Robot [C]. In Proceedings of 2008 IEEE International Conference on Networking, Sensing and Control. 2008: 938-942

[6] 岳基隆；四旋翼无人机自适应控制方法研究：硕士学位论文，国防科学技术大学研究生院，2010 年。

[7] Multiwii; http://www.multiwii.com

[8] 王俊，鲁晓天；对四轴飞行器基于姿态算法的分析与实现；《河南科技》2015 年 02 期。

[9] 戴连奎等；过程控制工程，第三版；化学工业出版社

[10] 刘浩蓬等；植保四轴飞行器的模糊 PID 控制；《农业工程学报》 2015 年 01 期。

[11] Peng, Tao; Sun, Lianying; Bao, Hong. Design and implementation of ATM simulation system based on MVC pattern. Educational and Information Technology (ICEIT), 2010 International Conference on , Issue Date: 17-19 Sept. 2010.

[12] 邱祖芳；基于 Android 平台的多点触控研究，硕士学位论文，南昌航空大学，2014 年。

[13] Sensors Overview| Android Developers; http://developer.android.com/guide/topics/sensors/sensors_overview.html

[14] 百度文库；Openwrt 路由器挂载摄像头教程。http://wenku.baidu.com/link?url=HTNBkeXQeAaX0tvntUIsiHpleJ1W8kvo0Bc3x63vBADnr75UAx3dOANR6jVX5h3ZuegJp-_2K7URTIudnyMyoN4k9qNeB8mnYjuyu

2oqEam

[15] Intel; USB 3.0 Radio Frequency Interference Impact on 2.4GHz Wireless Devices, White Paper, April 2012.

致谢

首先感谢我的导师卜佳俊教授，我的论文是在他的悉心指导下完成的。感谢他让我选择我感兴趣的课题来进行研究，并对我的论文进行审阅，指正。还帮助我规划毕业方向，我受益匪浅。

感谢我的班主任李铁风老师，他对我的研究提供了入门指导与器材支持，让我能够顺利地完成相关实验。

感谢我的辅导员吕成祯老师，他在大学四年里指导了我的生活和丰富的社团活动。

感谢我的朋友与家人。

本科生毕业论文（设计）任务书

一、题目：基于 WiFi 通信的四旋翼飞行器遥控系统设计

二、指导教师对毕业论文（设计）的进度安排及任务要求：

根据开题报告中提出的研究计划，学生要在四月初完成四旋翼飞行器的搭建，五月初完成手机 APP 的开发，并在六月上旬完成毕业论文撰写工作，完成研究计划中提出的基于 WiFi 通信的四旋翼飞行器遥控系统设计。

起讫日期 2015 年 1 月 5 日 至 2015 年 6 月 3 日

指导教师（签名）＿＿＿＿＿＿职称＿＿＿＿＿＿

三、系或研究所审核意见:

负责人（签名）＿＿＿＿＿＿

年　月　日

毕 业 论 文（设计） 考 核

一、指导教师对毕业论文（设计）的评语：

宋博同学在其所进行的毕业论设计“基于 WiFi 通信的四旋翼飞行器遥控系统设计”中，通过全面深入的项目调研，确定实施方案，并通过编程实现和大量实验，达到了导师所要求的基于 WiFi 通信的四旋翼飞行器遥控系统设计的要求。毕业设计语句通顺，逻辑严密，条理清楚，有很好的应用价值达到了本科生毕业设计的水平。

指导教师(签名）

年 月 日

二、答辩小组对毕业论文（设计）的答辩评语及总评成绩：

成绩比例	中期报告 占（10%）	开题报告 占（20%）	外文翻译 占（10%）	毕业论文（设计） 质量及答辩 占（60%）	总 评 成绩
分 值					

答辩小组负责人（签名）

年 月 日