

武汉工商学院毕业论文(设计)

学院：人工智能学院

专业：物联网工程 年级：2016 级

题目：基于 STM32 的无人机控制系统的设计与实现

2020 年 5 月 10 日

武汉工商学院

本科毕业论文（设计）原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：

2020 年 5 月 10 日

目录

摘要.....	I
关键词.....	I
Abstract.....	II
Key words.....	II
1 绪论.....	1
1.1 研究的目的和意义.....	1
1.2 国内外的主要现状.....	1
2 飞行控制系统总体设计方案.....	2
2.1 飞行器的组织结构.....	2
2.2 飞行器飞行原理.....	3
2.3 飞控系统的功能.....	5
2.4 飞控系统的总体设计.....	5
3 无人机飞控硬件分析.....	6
3.1 最小系统模块.....	7
3.2 传感器模块.....	7
3.3 电机驱动模块.....	9
3.4 通信单元模块.....	10
4 无人机飞控软件设计.....	11
4.1 软件开发环境介绍.....	11
4.2 飞控系统程序设计流程.....	12
4.3 无线通信.....	13
4.4 PWM 驱动电机.....	14
4.5 无人机姿态解算.....	15
4.6 PID 控制算法.....	18
5 无人机系统测试.....	20
5.1 通过串口对 NRF 配置.....	20
5.2 遥控器通道数据测试.....	22
5.3 传感器数据测试.....	23

5.4 PID 控制测试.....	25
5.5 飞行状态检测.....	26
6 结束语.....	27
参考文献.....	28

基于 STM32 的无人机控制系统的设计与实现

摘要

随着时代的进步，无人机在我们的生活中越来越普及，其中无人机涉及到的知识面也非常的广泛，涵盖了嵌入式，人工智能，自动化，通信等多个领域，非常适合将无人机作为载体学习更多和自己专业相关的知识。

本文主要实现的功能是通过遥控器的控制，实现飞机的运行轨迹的改变，控制四旋翼无人机运动方向，并且可以平稳降落，在紧急情况下也可以一键上锁，避免出现无人机伤人情况。

本文具体的介绍了开发一个完整飞控所需要的几个主要部分，包括飞控系统的总体设计，再根据主要硬件模块做详细的软件设计。系统主要使用 Keil 软件开发系统进行程序编写和调试，开发前期使用 STM32CubeMX 对需要硬件设备进行初始化。由于目前知识水平有限，不能搭载更多的模块实现对无人机更好地控制，后期会根据实际需求搭载更多模块提升无人机的整体性能。

关键词：嵌入式；无人机；飞控

Abstract

With the progress of The Times, uav is more and more popular in our life, and uav involves a wide range of knowledge, covering embedded, artificial intelligence, automation, communications and other fields, which is very suitable for uav as a carrier to learn more knowledge related to their own professional knowledge.

The main function of this paper is to change the flight path of the aircraft and control the movement direction of the four-rotor uav through the control of the remote control, and it can land smoothly. In case of emergency, it can also be locked with one key, so as to avoid the situation of killing people by the uav.

In this paper, the development of a complete flight control needs several main parts, including the overall design of the flight control system, and then according to the main hardware modules to do detailed software design. The system mainly USES Keil software development system for program writing and debugging. In the early stage of development, STM32CubeMX is used to initialize the required hardware devices. Due to the limited level of knowledge at present, more modules cannot be carried to achieve better control of the uav. Later, more modules will be carried to improve the overall performance of the uav according to the actual requirements.

Key words: Embedded; Unmanned aerial vehicle (uav); Flight control

1 绪论

1.1 研究的目的和意义

随着现代社会快速发展，高科技产品的应用迅速发展，无人机在大众的生活中也更加的普及，最近这些年很热门的研究就包括多旋翼无人机。它是拥有现代特别多高科技的成果，能够为工业、农业、现代军事提供非常多的便利。在农业方面，可以完成高空喷洒各种农药，自动播种；在工业方面，可以实现电力巡检，高空摄影，森林防火；军事方面是有实现灾区搜救，空中侦察。

无人机是无人驾驶、可以多次使用的一种动力航空器的简称。由于小型无人机的结构简单、稳定性高、价格低廉、低功耗得到了不少航模爱好者的喜爱，它的发展也是尤为迅速，具有广阔的发展前景。现如今的社会，对无人机性能相关要求也很高，无人机的稳定，安全，操作简单，数据实时性也成为了越来越多人困扰的问题。

作为新时代的一名优秀大学生，我也是对无人机很感兴趣。每个人都会有像鸟儿一样翱翔在天空的梦想，我也不例外，每每走在学校的操场上看着学长手中的无人机高高的飞在天空中，我就对无人机充满了期待与向往，梦想着自己也可以做一个无人机。所以本次的毕业论文设计让我有了一个设想，就是自己做一个简单的无人机出来，当然，四旋翼小型无人机的设计和本专业的嵌入式知识非常的契合，比如说用一个简单的芯片，构造出一个基本完善的飞行控制系统，实现无人机的驱动，这正是我们嵌入式所学的知识。综上所述：毕业设计优先考虑了无人机，做一个基于 STM32 系列的无人机控制系统。

1.2 国内外的主要现状

无人机的发展过程是缓慢的，也就是最近这几年才开始逐渐流行起来的。无人机的发展经历了好几个时期，首先是休眠期，也就是 1990 年以前，那个时候人们根本就不敢想除了鸟类和会飞的昆虫，还会有什么可以在天上飞；接着就是发展期，从上世纪 90 年代初，法国的 Breguet 兄弟开始了他们直升机飞行尝试开始，如下图 1-1 所示，到 1923 年 Etienne Oemichen 设计的直升机首次实现了在空中飞行 14 分钟的世界纪录，这个时候无人机才刚刚走进大众的视野，但是这个阶段的无人机基本是供与军方需求，技术上达不到实际的突破，于是转型开始走遥控玩具的市场，这才被大众了解的更加深入。然后就是活跃期，学者人数激增，都开始钻研多旋翼无人机，重点就是无人机的姿态结算和 PID 控制，也有

越来越多关于无人机的文章发表在各个著名书刊，其次就是越来越多关于无人机的开源项目面世，所有人对于无人机的学习是越来越方便，也可以在各个开源的论坛上进行相互之间的讨论。从 2013 开始，无人机行业就进入了爆发期，尤其是大疆 Phantom 的横空出世，引起了一场轩然大波，得到了全世界的持续关注，随后各个无人机的公司相继推出了各具特色的无人机，如美国的 3D Robotics 公司推出了 SOLO 无人机，瑞士的 Perspective Robotics AG 公司推出的 Fotokite Phi 无人机……

现如今国内无人机技术的发展也越来越迅速，我们日常生活中见到最多的就是航拍无人机，派送无人机。在今年这个特殊的日子，由于新型冠状病毒的影响，大家的生活也都有一定的不方便，这个时候无人机在很多的方面就发挥了比较大的作用。多个地区利用多旋翼无人机，安装扩音设备，可以实时向人民群众传递最新的防控消息，大力宣传防疫知识；利用无人机视野开阔看的東西多，视频传输高精度的特质，可以实时对多人聚集的情况进行反馈，达到减少群众聚集的情况；利用无人机可以跨区域操控的特性，在多旋翼无人机上悬挂二维码让群众扫描填写自己的个人健康信息，减少了相关人员的接触，减低了传染性。多旋翼无人机的运用，极大的提升了发现防疫问题处置的及时性和准确性，节省了疫情防疫工作的各种资源。

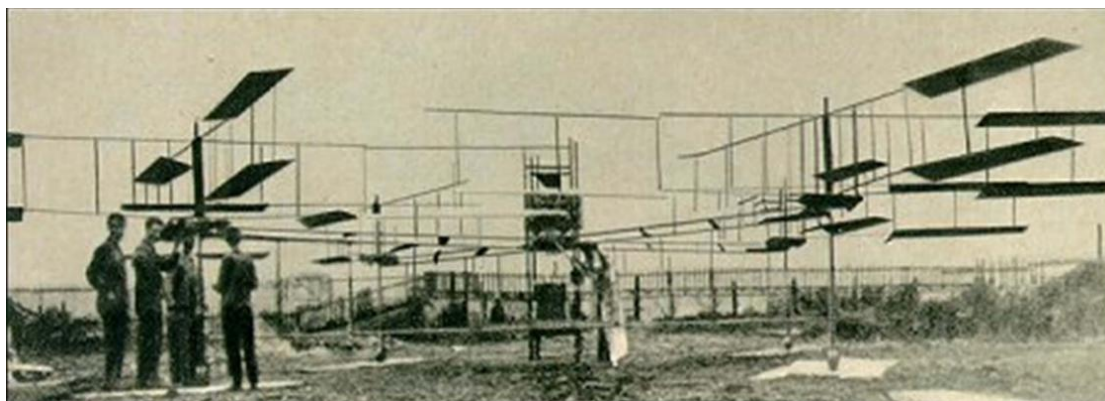


图 1-1 Breguet 兄弟创造的直升机

2 飞行控制系统总体设计方案

2.1 飞行器的组织结构

市面上通常见到最多的飞行器大致分为：固定翼，直升机，多旋翼。^[1]本次课程设计采用的是多旋翼的飞行器，伴有下面这些优点：

- (1) 能够垂直起降，不用发射架进行发射，可以在半空中悬停；
- (2) 机械构造较为简略，拆卸方便；
- (3) 成本低，易于维护；
- (4) 安全性高，控制灵活。

依照排列划分，有 3 旋翼，4 旋翼，6 旋翼，8 旋翼，12 旋翼等等；按照分布的位置进行划分，可以分为“+”型，“X”型，“H”型等等。课程设计选用的是 4 旋翼“X”型的无人机，它是最常见的一种类型，由于结构简单，飞行状态比较稳定，受到大部分航模爱好者的青睐。但是由于电机的功率受到制约，X4 在承载重量的方面不如 X6 与 X8，所以一半大型的航模和航拍无人机较多采用后两种形式作为自己的机架结构。

多旋翼无人机的硬件结构主要是分为机架、动力系统、飞行控制系统。机架又包括机身，机臂和起落架，机架将相当于人的躯干部分，提供一个大的框架；接着就是动力系统，动力系统有如下部分：电机；电调；螺旋桨；电池。动力系统就相当于人的血液，血液流动全身，带动身体的运转；飞行控制系统一般被人们简称为飞控，通常是由传感器、机载计算机和伺服动作设备组建。^[2]主要实现无人机姿态稳固和管制、无人机系统工作的设备管理和应急控制等，和人类的大脑类同，主动维持飞机非反常的飞行姿态就是它的主要功用。

2.2 飞行器飞行原理

在我们小的时候，科技没有那么发达，玩具也比较有限，相信肯定有不少人玩过“竹蜻蜓”这个玩具，当用人的双手迅猛的搓动“竹蜻蜓”下面的那根棒子，“竹蜻蜓”会迅速的旋转，从而产生一个升力让它起飞，如图 2-1 所示为“竹蜻蜓”示意图：

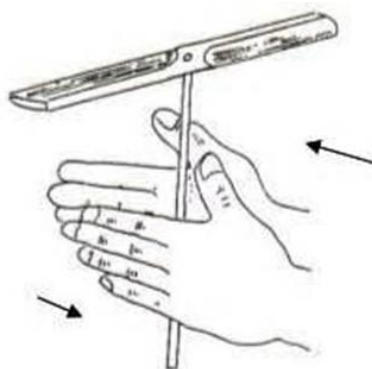


图 2-1 “竹蜻蜓”示意图

要是无人机转起来也差不多是这个理，无人机有四个电机，当电机迅速的扭转就可以带动螺旋桨的旋转，从而生出升力使得多旋翼无人机可以飞起来，当然飞起来的最重要条件就是螺旋桨的升力与无人机自身的重力相等。在无人机飞行过程中自身还会产生一个反作用力，致使飞机在测试飞行中会呈现出自旋，影响正常飞行，因此为了防止该类事件的发生，四旋翼无人机中螺旋桨紧邻的桨的旋转方向是不一样的。

如图 2-2 所示：红色的箭头表示四旋翼无人机的正方向，相当于机头方向，桨 M1 和 M3 体现为顺时针旋转，M2 和 M4 逆时针旋转；当在半空中飞行时，M1 和 M3 就会生出逆时针的反扭矩，M2 和 M4 造就的顺时针的反扭矩相互对消，飞机就可稳定飞行。当然，多旋翼无人机的运动状态，飞行的姿态，大多依靠多个螺旋桨的不同的转速控制来实现的。

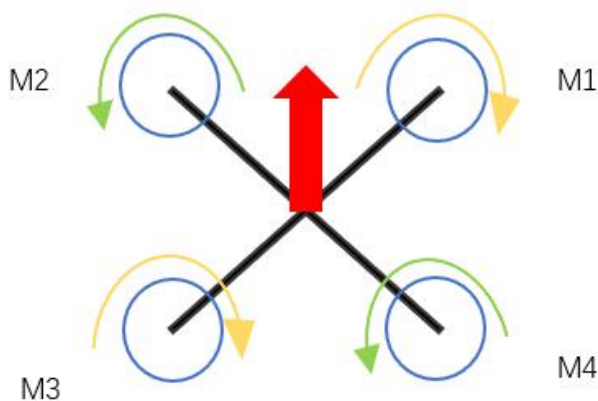


图 2-2 模拟飞行器

根据伯努利原理，流速大的压强小，流速小的压强大。^[3]对螺旋桨查看剖面分析，上表面比下表面面积大，单位时间内，空气流经下表面的速度大于上表面，由此两表面显现的压力差就会促使桨叶出现升力，如图 2-3 所示：

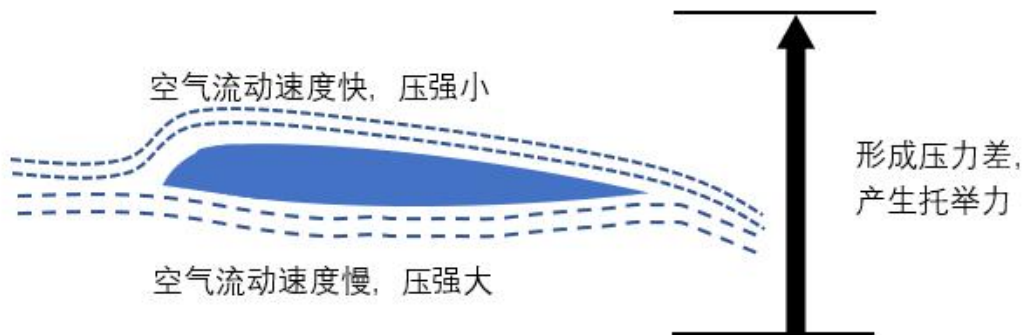


图 2-3 模拟伯努利原理图

依照空气动力学可知，螺旋桨上产生的升力与反转矩、空气密度、螺旋桨的速度长度、桨叶的特征升力系数等都相关。螺旋桨形成的动力是导致多个转子 UAV 运动的主要提升力。螺旋桨的动态特性对多旋翼无人机的系统模型和运动控制相当的重要，不同的机动可被无人机执行。

2.3 飞控系统的功能

在无人机的组织结构中，飞控无疑是最重要的一部分，它控制着无人机的飞行姿态，运动轨迹，并且可以根据指令做出相应的改变。飞控系统具备的主要功能有：

- (1) 飞行轨道的改变，利用拨动遥控器完成对飞控的控制，管制四旋翼无人机的飞行轨迹；
- (2) 平稳降落，通过遥控器促使四旋翼无人机四个电机的油门得到控制，达到飞机平稳降落的状态；
- (3) 一键上锁，在紧急情况下飞控不受控制，可以通过遥控器实现紧急上锁，避免出现无人机伤人情况。

2.4 飞控系统的总体设计

无人机在现代生活中越来越普及，完成的工作也更繁琐，从简单的遥控控制到如今的智能飞行，无人机的发展突飞猛进，当然也少不了对飞控的高要求，如反应灵敏，飞行状态稳定，低功耗，低成本。飞控在多旋翼无人机中是十分关键

的,没有飞控的控制无人机就不存在飞行,所以飞控的设计要结合多方面的因素,才能保证最优的飞行。

系统利用 C 语言进行编程,单片机 C 语言软件开发系统 Keil 作为主要开发工具,采用 STM32F103 作为系统的主控芯片,该芯片以最小的硬件变化满足无人机的多种应用需求。系统设计有两个输入,一个输出,首先通过接收机接入遥控器发来的顶层数据,控制无人机是往前飞多少还是往右飞多少;接着就是传感器给的姿态输入,控制无人机的姿态变化,遥控器的数据通过无线通信模块 NRF 将数据传输给到主控芯片,传感器通过 IIC 总线传输给到主控芯片,主控芯片最后通过控制电机做输出,也就是利用 PWM 信号控制当前电压的百分比达到对几个电机的控制,由此无人机即可实现飞行,同时通过开关按键控制飞控电源的输入,利用不同的 LED 灯闪烁的情况控制当前飞控的状态,主要的飞控系统设计如图 2-4 所示:

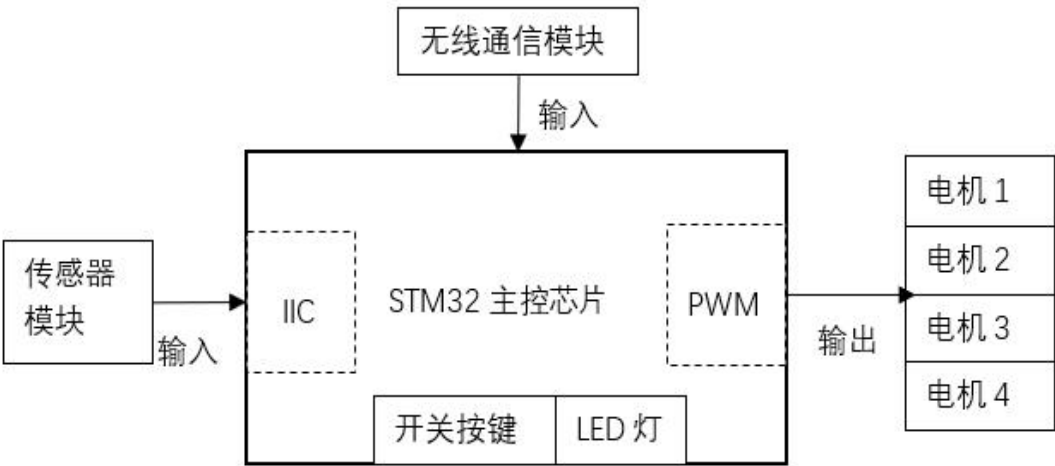


图 2-4 飞控系统设计结构图

3 无人机飞控硬件分析

无人机飞控开发平台硬件组成有：飞控核心、电机，螺旋桨、传感器模块 F230 机架。飞控核心主要由：中央处理器 STM32F103C8T6 主控芯片、电机驱动、电源管理、无线通信模块、姿态传感器以及气压计组成。

- (1) STM32F103C8T6——中央处理器，控制最重要的部分；
- (2) 电机驱动——是直流无刷电机的驱动单元；
- (3) 电源管理——电压变换以及电池充放电管理；
- (4) 无线通信模块——负责和遥控器的双向通信；
- (5) 姿态传感器——是两个传感器：有陀螺仪、加速度计，输出无人机的姿态；

3.1 最小系统模块

系统采纳的是将 STM32F103C8T6 作为主控芯片，这是一款基于 ARM Cortex-M3 作为内核的 32 位的微控制器。^[4]系统存储器容量为 64KB，必要的工作电压是 2V-3.6V，稳定的运行频率能够到达 72MHz，具备 12 位模数转换器，7 通道的 DMA 控制器，3 个通用计时器和 1 个高级控制计时器，37 个快速输入输出端口，串行线调试器(SWD)和 JTAG 接口，2 个 IIC 接口，3 个 USART 接口， 2 个 SPI 接口，一个简约的飞控系统的设计它都可以满足。

该主控芯片存在高性能、低功耗、低电压、低成本的特点，而且对中断有更好地响应，这款主控芯片强大的软件支持也提供了不少便利，具有丰富的技术文档和强大的用户基础，为我后来设计提供了帮助，如图 3-1 所示为 STM32F103C8T6 最小系统原理图：

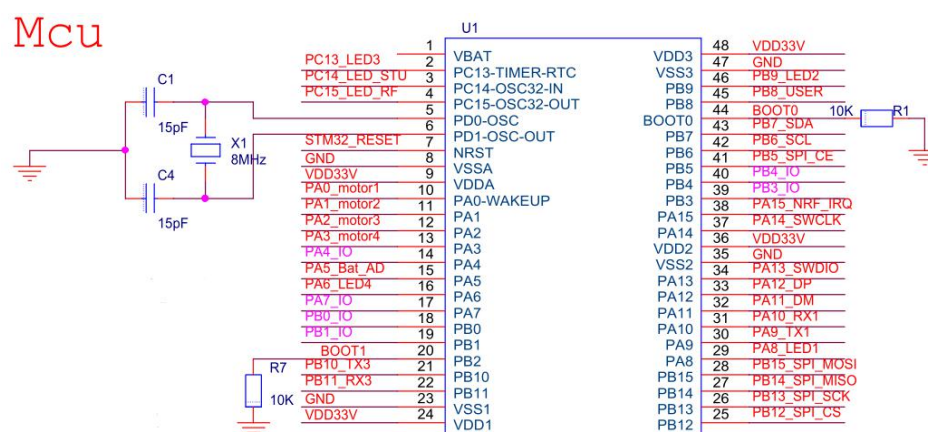


图 3-1 STM32F103C8T6 最小系统原理图

3.2 传感器模块

系统取用 MPU6050 当作姿态传感器，用于收集有关飞行过程中用于执行实时电机控制的位置的数据，是种位置传感器，为了识别姿态传感器 MPU6050 的工作原理，首先需要了解两个传感器：陀螺仪和加速度计，其实 MPU6050 传感器就是由这两个传感器组成。^[5]

（1）陀螺仪

如图 3-2 所示是模拟的陀螺仪内部的一个原理图，陀螺的原理：旋转对象的旋转轴的方向在不接受外力的情况下不会变动；一旦受到外力的影响，他会在外力的垂直方向上有一个反向作用力，来尽可能让这个陀螺保持稳定，保持平衡。日常生活中骑自行车时，轮儿转得快就不易摔倒，由于车轴有一股维持水平的力量。



图 3-2 陀螺仪内部原理图

MPU6050 传感器的内部蕴含一个结构精细的陀螺仪。旋转速度和角度可以通过外部三个圆（代表三个方向）和初始方向之间的偏差来计算。始终保持平衡其实就是说这个金属轴，刚开始是什么样，在运动的时候也是什么样，不会发生任何改变。简而言之，陀螺仪是一种隐式测量角度的装置，它测量角度的导数，即角速度，只有当角速度对时间积分，这时有可能得到一个角度。

从理论上讲，陀螺仪可以执行姿态导航任务的，只需将陀螺仪角速度积分集成到三个轴中，即可获得三个方向的旋转角度，即所谓的姿态数据。因为误差、噪声等的存在，比如说出厂的时候肯定会有部分误差，运行的时候也会有信号干扰，生成误差，陀螺仪就慢慢积分，因此得不到完全精准的姿态。特别是在一段时间的操作之后，积分错误的积累使得这种姿态远离了实际。积分时长越大，近似值误差就会越大，这个时候我们光有一个陀螺仪是不够的，它不能够完整的描述一个运动姿态，所以我们需要添加加速度计传感器来帮助控制陀螺仪收集的数据，以便数据更准确。

(2) 加速度计

加速度计用于测量静态加速度，假定它们用于飞机，实际就是对重力加速度 G 的测量和剖析。



图 3-3 无人机运动状态模型图

假设有一个无人机实时运动状态模型，如图 3-3 所示：这个图的左边可以理解为无人机处于平衡状态下的一个加速度计，在垂直方向上有 $1g$ 的重力分量，在水平方向上是没有任何分量的，一旦无人机的状态发生改变，角度自然就会发生变化，就像这个图的右边所示，可以看到重力加速度在水平方向上也有一个分量，这个分量其实是用 $1g \times \sin\theta$ 算出来的，利用三角函数的知识，就可以得到这个角度值，这就是加速度计的原理。

系统采用了 IIC 数字接口，一根数据线 SCL 时钟线连接 STM32 的 PB7_SDA 引脚，另外一根 SDA 数据线连接 STM32 的 PB6_SCL 引脚，C2、C3 作为电压上进行滤波使用，通过 IIC 总线给出六个维度的 ADC 值，同时自己提供一个从 IIC 接口，初始化 MPU6050 时，设置成主 IIC 总线与从 IIC 总线直通，STM32F103C8T6 可以直接通过主 IIC 总线访问 IIC 总线，从而获得 MPU6050 数据，如图 3-4 所示为 MPU6050 电路原理图：

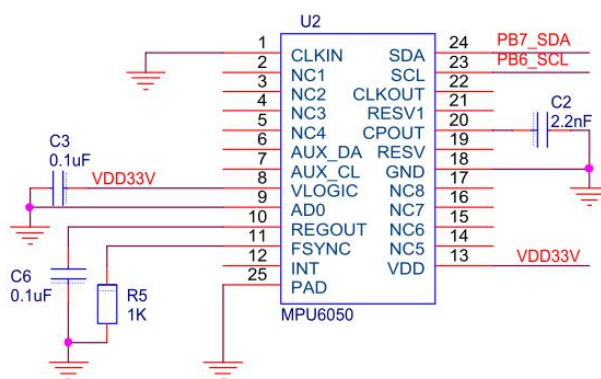


图 3-4 MPU6050 电路原理图

3.3 电机驱动模块

多旋翼无人机动力维持是从螺旋桨处获得的，这是发动机旋转的结果。无人

机行业中，小型无人机通常运用有刷电机，如空心杯发动机，而大轴底座若逾越 200 毫米，则利用无刷电机。该系统的无人机使用 230 毫米的轴底座，选用的是无刷电机。

无人机框架系统配备了高速 MOS 管，可直接驱动无刷电机。PWM 信号的输出用于调整 MOS 管的快速通断，以更改输出的平均电压来调整发动机转速和扭矩。要更改无刷电机的旋转方向，只需更改发动机动力源的正极或者负极即可。如图 3-5 所示，为电机驱动电路原理图。

原理图由 MOS 管 Q1、多个电阻、电机 M2 和电池组成；电机 M2 的一端连接到 Q1 MOS 管的漏极，另一端与电池正极相连；MOS 管门极与单片机输出 PWM 信号的引脚相连即 PA2_motor3，并且并联 10K 欧姆的电阻下拉到电源地，MOS 源极一同连接到电源地，当系统通电以后，电流由电池正极流向电机，并通向 Q1 MOS 管，若 MOS 管导通，则电流顺利流经 MOS 管后回流至电池，构成闭合回路，电机即可顺利转动。

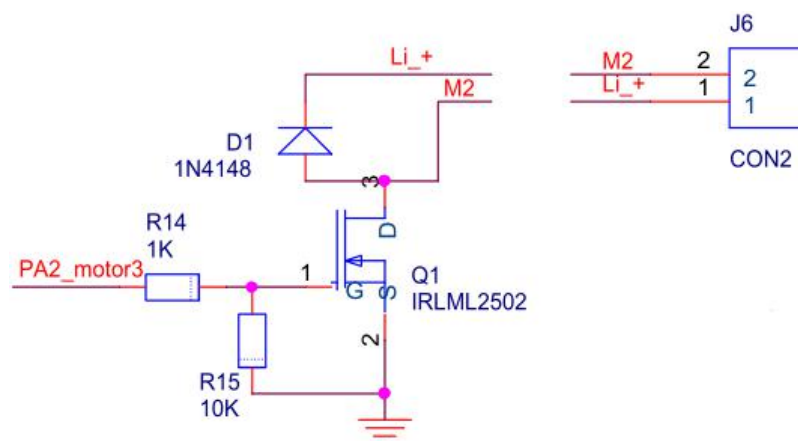


图 3-5 电机驱动电路原理图

3.4 通信单元模块

现在无人机的通用射频为 2.4GHz，该项技术已成为无线传输的通用技术，ZigBee 也是基于 2.4GHz 的，然而它运用不同的路由方法会有不一样的传输速率，因此流量跨度也不同，使用相同频率作为传输办法，但是来自不同通信方法的通讯方式将会大不相同。

系统采用 nRF24L01 芯片用作遥控器和飞控之间的通信方式，能够做到点对点、1 对 6 的无线通讯，速度达到 2M (bps)。软件程序的编写也不复杂，仅操作 5 个 GPIO 引脚，1 个中断输入引脚，就能够实现无线通讯，无人机飞控核心以及飞控开发平台的遥控器都基于 nRF24L01 设计了双向无线通信电路，如图 3-6 所

示：

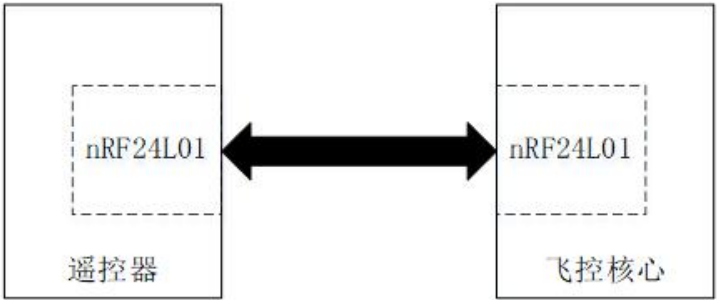
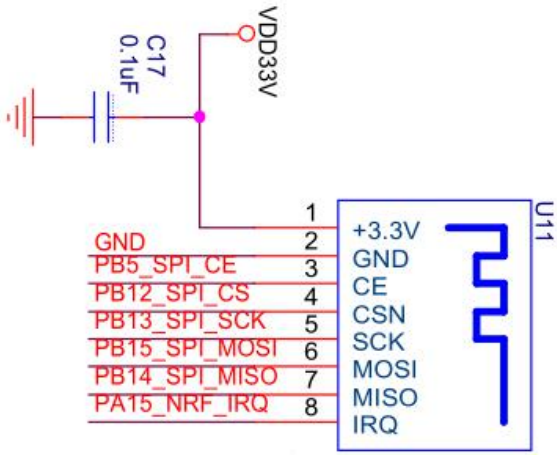


图 3-6 无线通信结构图

nRF24L01 与控制器的通信采用的 SPI 通信协议，单片机控制只需要关注这几个引脚：CSN、SCK、MISO、MOSI、IRQ、CE。



飞控核心的无线通信模块电路原理图如图 3-7 所示：

图 3-7 无线通信模块电路原理图

4 无人机飞控软件设计

4.1 软件开发环境介绍

Keil 是软件开发系统，源自于美国，可以进行微控制器的开发，能够实现汇编语言、C 语言的编程，并且提供了一个功能完善的仿真调试器，拥有一个强大

的集成开发环境，可以充分完成对嵌入式的开发应用。Keil 根据所选芯片种类的不同提供了多种库函数，极大的提高了使用者的使用效率，可以很方便的对源代码进行编译链接，调试和编译报错也给使用者提供便利，Keil 开发环境界面如图所示：

STM32CubeMX 是最近这些年 ST 公司推出的 STM32 系列芯片图形化的一个配置工具，大大减少了使用者工作量，它能够快速的创建工程，通过图形化的界面可以对我们硬件设备进行初始化，并对外设做抽象化的处理，让开发者更多的把时间用在应用层的开发，STM32CubeMX 开发环境界面如图 4-1 所示：

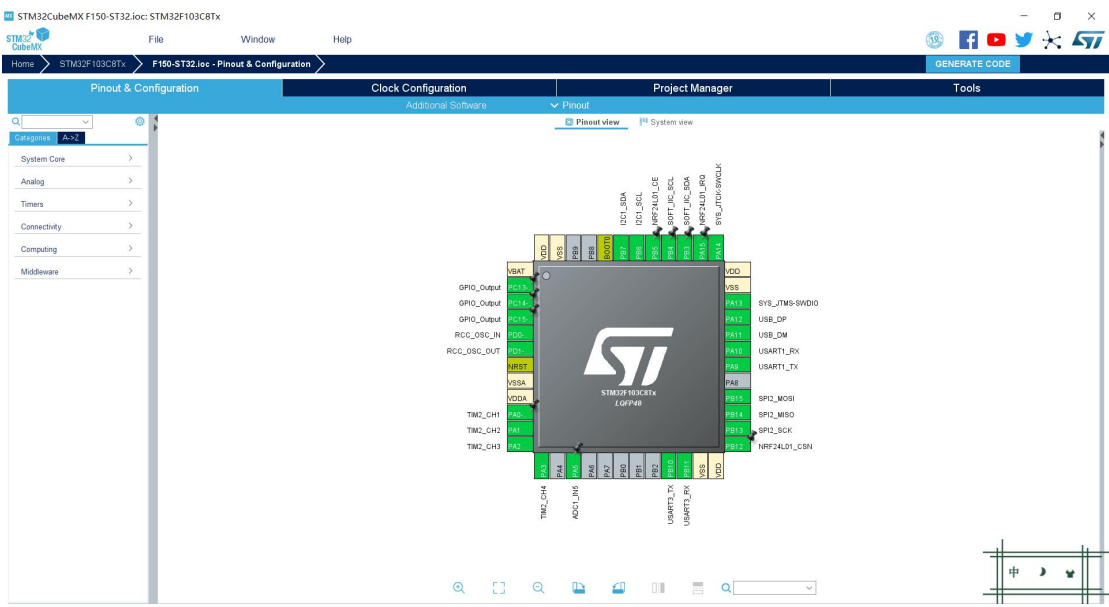


图 4-1 STM32CubeMX 开发环境界面图

4.2 飞控系统程序设计流程

飞控系统主要用来保持无人机平稳飞行，是无人机的最基本的点，系统飞控选取 STM32F103 芯片作为处理器，整个飞控系统依靠定时器中断处理优先级不同的各项任务，利用 MPU6050 传感器作为惯性测量单元得到原始数据，根据原始数据计算出无人机当前姿态，再联合遥控器输出的目标值数据，两者的差值再经过 PID 控制算法驱动无人机电机转速来达到无人机按照遥控要求实现稳定飞行。系统的主要设计过程是先启动微控制器，然后启动整个飞行控制系统，包括 PWM 启动，GPIO 启动，串行检查启动，传感器启动，PID 启动，整个系统初始化完成以后就会进入到核心轮训，等待遥控器的指令，飞控得到遥控器的指令后就开始读取 MPU6050、气压计的数据，对读取到的数据做姿态解算，解算完成以后通过闭环 PID 进行电机控制，最后将输出的值通过 PWM 实现对电机的调速，达到飞机

平稳的飞行，如图 4-2 所示为系统程序设计流程图：

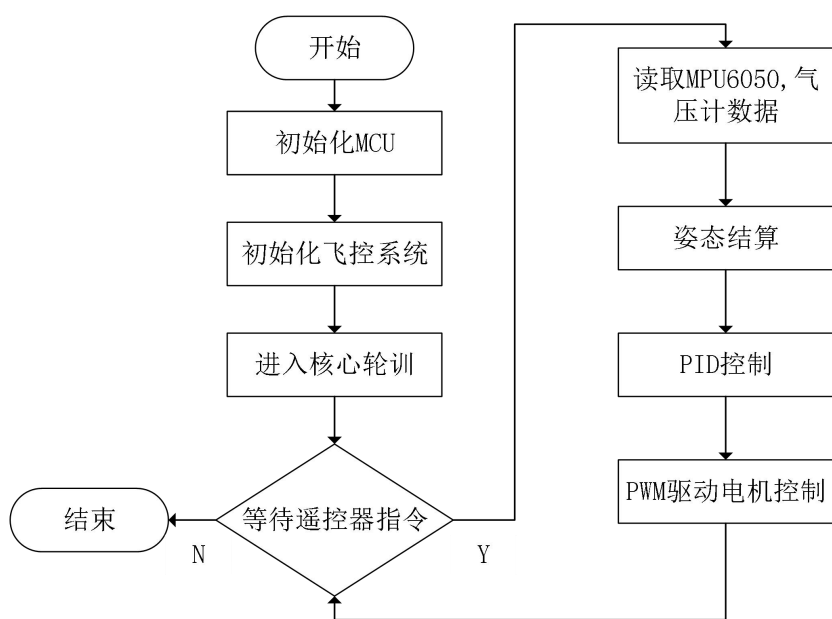


图 4-2 系统程序设计流程图

4.3 无线通信

无人机和遥控器之间的通讯是基于 nRF24L01 模块，该模块应用 SPI 总线与 MCU 构成通信。完全相同的 nRF24L01 单元同时插入到遥控器和无人机核心中，遥控器中的 NRF24L01 单元配置为传输形式，无人机核心中的 nRF24L01 单元设置为接收器形式，这样遥控器就可以向飞控核心发送指令，遥控器和飞控核心任何一个设置为接收或者发送模式都是可以的，然后飞控核心就能接受到遥控器的指令，通过遥控器的各种指令控制无人机飞行状态和运动轨迹。

每个 nRF24L01 设施都有一个接收地址和多个发送地址，根据接收设备的接收地址和频率就可发送有用信息至相应设备，每个地址由 5 个 byte 组成，可接收和发送的长度最大限制为 32 个 byte。

要实现无线通信首先要对 nRF24L01 模块进行初始化，nrf24l01_init() 函数首先在 while 中做判断，判断是否能够通信成功，通过调用 check() 函数，查验 nRF24L01 硬件是否链接成功，然后实现数据的读写操作，接着进行寄存器的配置，第一步需要写发送节点地址，第二步写接收节点地址，这两步都是在 WriteBuf() 函数中实现的，这两个步骤主要是将读取的数据写入这两个寄存器中，然后启用自动响应通道 0，激活相应的通道地址 0，设置自动重传周期，最大自动重传次数为 10，并将 RF 信道频率设置为 40，设置射频寄存器，以上都是

通过 WriteReg() 函数实现，初始化完成以后需要通过 RxPacket() 函数判断当前状态是否数据接收完成，若接收成功状态值就返回 true，set_nrf24l01_rxaddr() 这个函数实现更新 nRF24L01 的接收地址，先禁止 nrf 接收模式 CLR_NRF24L01_CE；调用 WriteBuf() 函数，批量写入 RX_ADDR_P0 通道寄存器数据，然后调用 ReadBuf() 函数再读取 RX_ADDR_P0 寄存器写入的地址，最后判断数据更改是否成功，成功就状态值就返回 true，最终按照接收到的指令完成对飞控的控制，如图 4-3 所示为 nRF24L01 无线通信流程图：

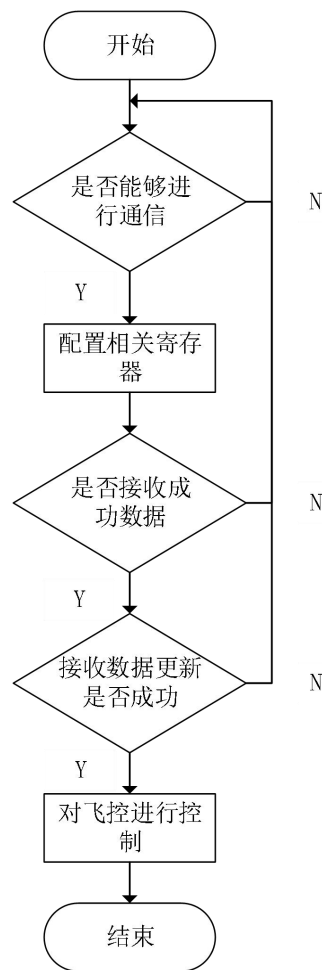


图 4-3 nRF24L01 无线通信流程图

4.4 PWM 驱动电机

STM32F103 系列单片机内部存在 11 个定时器，定时器单元主动迸发 PWM 输出，用于产生控制信号。一个 PWM 的周期等于迸发 PWM 的定时器的周期。因此，仅需一个带有可调节操作环的 PWM 信号即可检测电动机速度设置，电机上的平均压力越大，速度越高，反之亦然。PWM 的初始化首先要对系统指定的时钟进行初始化，

系统需用的是 TIM2 时钟，时钟频率定为 1000Hz，通过 TIM2 设置 4 个 PWM 通道，如下给出了部分电机初始化的代码：

```
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);  
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2 );  
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_3);  
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_4);
```

初始化定时器结束后，接着输出 PWM 信号掌控电机转速了，系统通过 HAL_PWM_DutyChange（）函数控制输出 PWM 信号的占空比；接着开始设置定时器 CCR 寄存器的值从而改变占空比，这样就可以掌控电机转速的同时管制飞行姿态。如图 4-4 所示为电机驱动流程图：

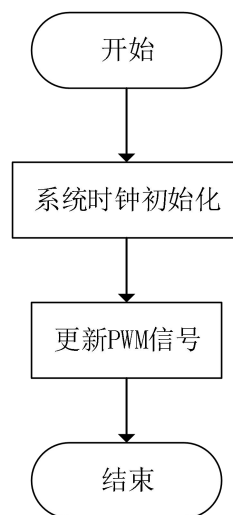


图 4-4 电机驱动流程图

4.5 无人机姿态解算

加速度和角速度信息设置在 0x3B-0x48 这 14 个字节的寄存器中，并且该信息将以可移动的方式更新，最大刷新速率为 1000 Hz，如图 4-5 所示，给出了姿态数据相关寄存器的地址及数据意义，每个数据具是 2 个字节。

地址	含义
0x3B	加速度计的 X 轴分量 ACC_X
0x3D	加速度计的 Y 轴分量 ACC_Y
0x3F	加速度计的 Z 轴分量 ACC_Z
0x41	当前温度 TEMP
0x43	绕 X 轴旋转的角速度 GYR_X
0x45	绕 Y 轴旋转的角速度 GYR_Y
0x47	绕 Z 轴旋转的角速度 GYR_Z

图 4-5 寄存器地址数据图

MPU6050 芯片坐标系是这样规定的，将芯片系统整体表面向上，以人们看到的正向旋转该芯片，以其核心为原点，水平向右的标为目的为 X 轴，竖直向上的标为目的为 Y 轴，指向自身的方位为 Z 轴，^[6]如图 4-6 所示：

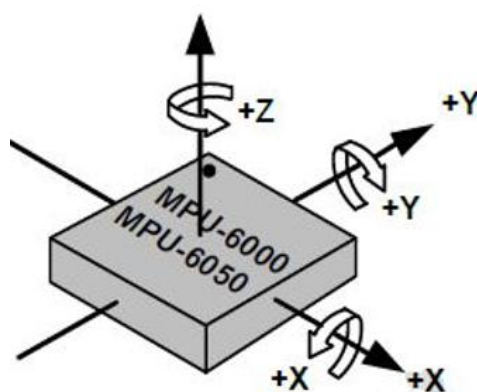


图 4-6 MPU6050 坐标系定义图

(1) 加速度计的三个轴的分量各自取名为 Acc_x、Acc_y 和 Acc_z，其中都是 16 位有符号整数，可以区分表现器件在三个轴方向上的加速度，如果为正值，则沿着三个轴的任一正方向，如果为负值，则是沿着三个轴的任一负方向。三个速度分量均基于失重数 g，该失重数 g 用于表示加速度范围。

(2) 角速度分量 Gry_x、Gry_y 和 Gry_z 绕 X、Y 和 Z 轴旋转，同为 16 位有符号整数。以核心为中心向其旋转轴方向望去，为正则表明顺时针旋转，三个角速度均为度/秒，能够表明角速度范围，则可以平均设置倍率。

无人机的姿态解算主要是依靠初始化和姿态解算。首先要对 IIC 总线设置一

个初始化,利用 IIC 总线采集得到 MPU6050 的数据;接着初始化 MPU6050 传感器, MPU6050 初始化其实就是对 MPU6050 传感器中的寄存器操作,写入相干的值,去确定相应的功能,置陀螺仪采样频率为 1000Hz,设置时钟源和陀螺仪 Z 轴,配置低通滤波频率 42Hz、配置加速度和陀螺仪量程为 $\pm 4G$ 和 $\pm 2000\text{deg/s}$;;

```
HAL_I2C_Write_Byte(MPU6050_ADDRESS, SMPLRT_DIV, 0x00);
```

```
HAL_I2C_Write_Byte(MPU6050_ADDRESS, PWR_MGMT_1, 0x03);
```

```
HAL_I2C_Write_Byte(MPU6050_ADDRESS, CONFIGL, 0x04);
```

```
HAL_I2C_Write_Byte(MPU6050_ADDRESS, GYRO_CONFIG, 0x18);
```

```
HAL_I2C_Write_Byte(MPU6050_ADDRESS, ACCEL_CONFIG, 0x18);
```

然后校准 MPU6050 的数据,传感器初始化完成以后,通过 update_mpu6050 函数读取陀螺仪和加速度计的数据并做滤波处理;分别对加速度数据进行一维卡尔曼滤波,对角速度数据进行一阶低通滤波,然后即可进行姿态解算,姿态解算是在 imu.c 文件实现的,执行 update_attitude_quad() 函数后,根据传感器数据计算得到当前姿态,姿态数据主要由欧拉角的样式展现,如图 4-7 所示为无人机姿态解算流程图:

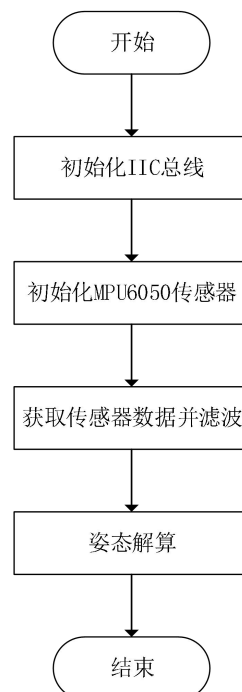


图 4-7 无人机姿态解算流程图

4.6 PID 控制算法

PID 控制器在工业控制中属于通用的一种闭环控制系统，该控制器具备两个输入，一个输出，两个输入包括人们操作时的目标值和飞机飞行中的测量值，依照两者的差值计算出新的输出值，这个新的输出值就可以使系统保持目标值附近，这样可以是系统保持在一个更稳定的状态。PID 管制的取得的“P”对应的是当时误差，“I”对应的是过去时刻累计的误差，“D”对应的是未来时刻的误差，如果不清楚受控系统的特点，基础 PID 控制器是被人们以为最理想的一种控制器，根据 PID 的三个参数可以实现对系统的控制，充分满足对系统的设计需求。

PID 控制器从 MPU6050 传感器拿到测量结果，接着用需求数据减去测量数据从而得到误差，然后采用误差来计算纠正值当作输入数据，这样系统就可以从它的输出结果中取消误差。如果是将 $u(t)$ 定义为 PID 控制输出，那么 PID 的控制算法可以表示为：

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

在本式中 K_p 作为比例增益， K_i 作为比例增益， K_d 作为比例增益， e 作为为误差，也就是由设定的值减去反馈的值， t 就是当前时间，作为一个积分变数，从 0 一直计数到 t 。PID 控制器广泛的被人们当作为频域系统的一个滤波器，大部分情况计算控制器能否达到一个稳定的状态，是评判系统好坏的依据，一旦数值筛选失误，控制系统的输入数据就会呈现不断震荡，因此系统永远不可能达到设定值的稳定状态。

PID 的控制算法位于 `pid.c` 和 `pid.h` 文件中，系统使用的是一般 PID 算法，可以增加积分限幅和输出限幅功能。PID 的控制算法首先会初始化其部分参数，然后开始计算角度环输出量，将角度环输出量作为速度环输出量的输入，再求解速度环的输出量，接着就能得到电机输出量。无人机总共有 3 个轴需要进行 PID 计算，所以实际的计算应是此处的三倍，当 PID 初始化完成以后，就可以开始循环计算 3 轴输出值。

飞控内 PID 控制算法的核心代码，以 `update_pid()` 函数为主，参数中包含一个 `pidinfo_t* pid` 要计算的 PID 结构体指针，其中具有一般 PID 控制算法中需用到的众多变量。`update_pid()` 函数是 PID 控制算法的核心，该函数主要负责依照 PID 公式，更新 PID 输出量，并更新结构体变量。`update_pid()` 函数会先计算

角度输出值，然后将角速率的目标值设定为角度的输出值，计算角速率输出值后得到电机的输出值，通过 `claccascade_pid()` 函数实现串级 PID 思想，在实际运行中，能够绵延不绝的迭代 3 个轴的 PID 输出量，如图 4-8 所示为 PID 控制算法的流程图：

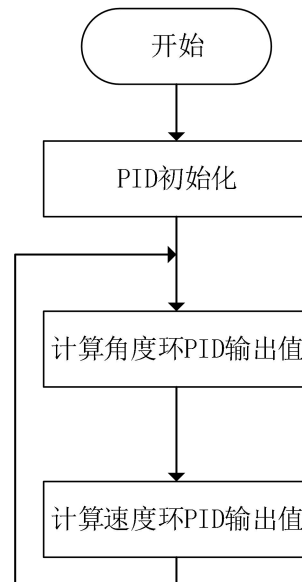


图 4-8 PID 控制算法的流程图

5 无人机系统测试

一个完整的飞控程序写完之后是不可以马上进行飞行测试的，因为无法验证程序的好坏和飞行过程中的稳定性，这会招致飞机在起飞过程中炸机，浪费人力和物力资源。于是便有了通过上位机对程序的好坏做测试，该无人机动态调参上位机系统是基于 Winform 开发环境进行的，通过串口的连接，可以实现对无人机各种参数配置的获取，也可以调节飞控和遥控器的收发地址、收发频率，对飞控 PID 数据进行调整和曲线显示。

5.1 通过串口对 NRF 配置

当飞控通过串口与上位机连接以后，飞控会将自己的默认地址，默认频率，上电时间发送给上位机显示，证明当前飞控状态良好。接着就需要对飞控和遥控器进行 NRF 的收发地址和 NRF 的收发频率配置，只有当两者的收发地址和收发频率一致时，飞控和遥控器才能实现正常的通信，遥控器才能够实现对飞控的控制。若要对 NRF 的配置，需要进入到主界面的“飞控 NRF 配置”一栏进行配置，输入新设置的 NRF 地址和 NRF 频率，设置完成以后点击确定按钮，上位机会将新设置的 NRF 地址和 NRF 频率存放在字符数组中，且频率设置仅为 0-64，然后通过串口将存放的字符数组都发送给飞控，飞控接收到串口发送过去的的数据就会进行频率和地址修改，为了保证地址的准确无误，飞控会在修改完成之后再向上位机返回一个修改完成的 NRF 地址和 NRF 频率，如果设定的 NRF 地址和 NRF 频率与返回的相同，则说明设置成功。遥控器的 NRF 地址和 NRF 频率也要和飞控的设置一样，当两者都设置完成则可正常通信，整体测试流程如图 5-1 所示，测试界面如图 5-2 所示，图片左下角位置则为修改成功的飞控 NRF 地址和 NRF 频率：

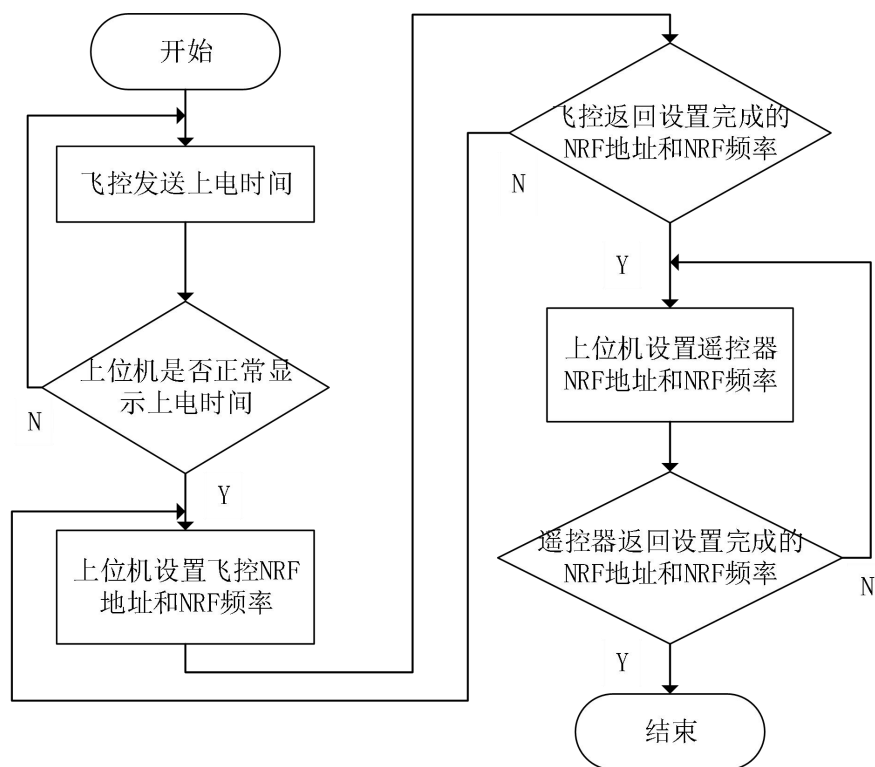


图 5-1 NRF 配置流程图



图 5-2 NRF 测试图

5.2 遥控器通道数据测试

要想实现飞控的飞行，必须通过遥控器的油门和方向控制才行，上一步已经配置好飞控和遥控器的 NRF 地址和 NRF 频率，接下来就可以利用遥控器油门和方向的调整控制飞控的飞行轨迹和运动状态了。通信模块会根据遥控器当前的输入量以固定长度发送遥控器当前的信息到上位机，遥控器上的每个控制量都是依照飞控通用标准设计的，数值范围在 1000-2000，例如摇杆最左边的输出量为 1000，最右边的输出量为 2000。通信过程中遥控器会发送一个固定形式的消息，起始位固定为 0xAA，确定数据长度为 0xC，发送 ID 为 02，接收 ID 为 00，消息 ID 为 01 和具体的数据，最后将以上需要发送的所有数据进行结算得到总和，需要以结构体的形式封装成为一个数据包发送给我们的上位机，上位机接收数据并解析，解析成功之后就会显示在界面上，对应的遥控器状态界面的遥控器通道数据下面的绿色方块会随遥控器的拨动而变化。整体测试流程如图 5-3 所示，测试界面如图 5-4 所示，通道数据下面的绿色方块则代表的遥控器拨动位置，如当左右拨动遥控器右侧摇杆时，“Roll”字样右边的绿块会跟随拨动的大小浮动，当遥控器右侧摇杆拨到最上面的时候，“Pitch”字样右边的绿块会占满空白区域，当遥控器右侧摇杆拨到最下面的时候，“Pitch”字样右边的绿块会消失，如图 5-4 所示将遥控器右侧摇杆往右拨动了一点：

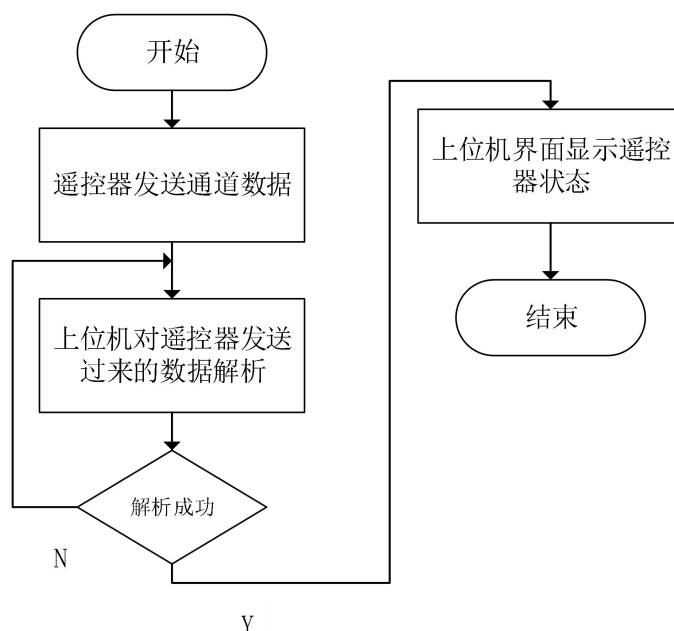


图 5-3 遥控器通道数据测试流程图



图 5-4 遥控器通道数据测试图

5.3 传感器数据测试

通过串口连接上飞控以后，飞控的传感器数据会实时将数据直接发送给上位机，上位机解析飞控发过来的数据，解析完成之后直接通过主界面飞控数据区的 IMU 数据进行显示，IMU 数据的前六个数据分别为加速度计和陀螺仪 X, Y, Z 三个轴的数据，这三个轴的数据为-32768 到32768，当测试数据超出这个范围说明传感器采集数据出错，当某一个轴的数据一直增大或者缩小时，说明姿态结算没有写成功，飞机在飞行过程中会出现漂移情况，然后就需要回到我们的源程序检查我们代码的准确性，修改完成，再查看上位机发来的数据无误，说明传感器数据正常。整体测试流程如图 5-5 所示，测试界面如图 5-6 所示，图片左下角位置“IMU 数据”中则为加速度计和陀螺仪的数据：

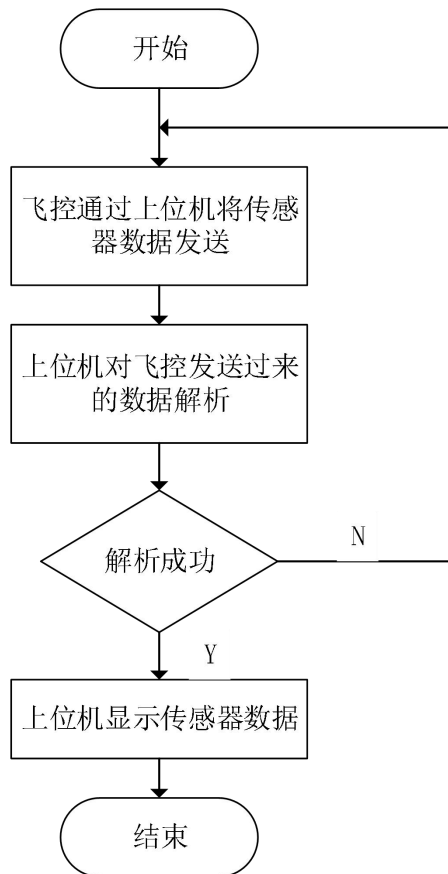


图 5-5 传感器数据测试流程图



图 5-6 传感器数据测试图

5.4 PID 控制测试

通过串口连接上飞控以后,通过上位选择当前控制器来查看具体的PID 数值,对飞控数据 PID 进行调试,通过“选择当前控制器”字样右边的方框选择具体的控制器查看。选择后,需要单击“确定”按钮。上位机将有关特定控制器的信息发送给飞控。飞控从上位机接收到信息后,会将上位计算机选择的 PID 飞控数据,发送到上位计算机,上位机解析,解析成功之后就会将数据通过曲线的形式进行展示,分别为测量值,目标值,误差值,当测量值与误差值之间的差距趋于零的时候说明无人机的飞行状态趋于稳定,可以用来飞行。如图 5-8 所示:如果需要实时调节 PID 参数值,可以通过左下方的[参数整定区]对 PID 的数值进行修改,修改的流程也是上位机发数据给飞控,飞控返回数据给上位机显示。整体测试流程如图 5-7 所示,测试界面如图 5-8 所示,当“Kp” “Ki” “Kd” 的值分别为 1.50, 1.10, 0.04 时测量值曲线和目标值曲线接近重合,说明飞行状态最稳定, PID 的值设置的最好。

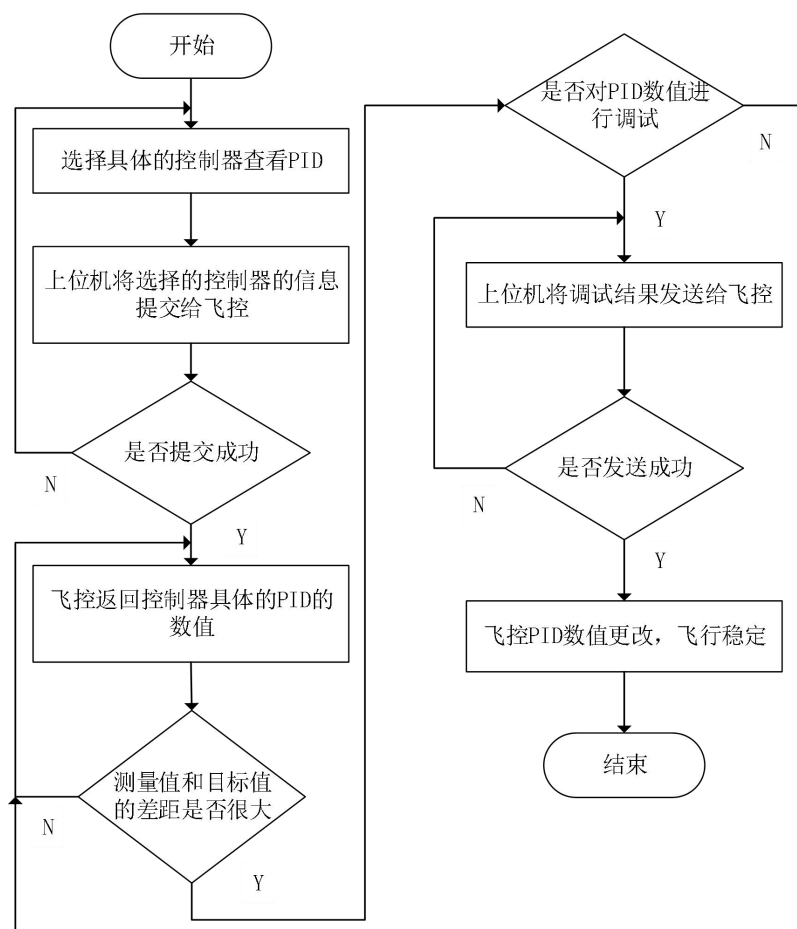


图 5-7 PID 数据测试流程图

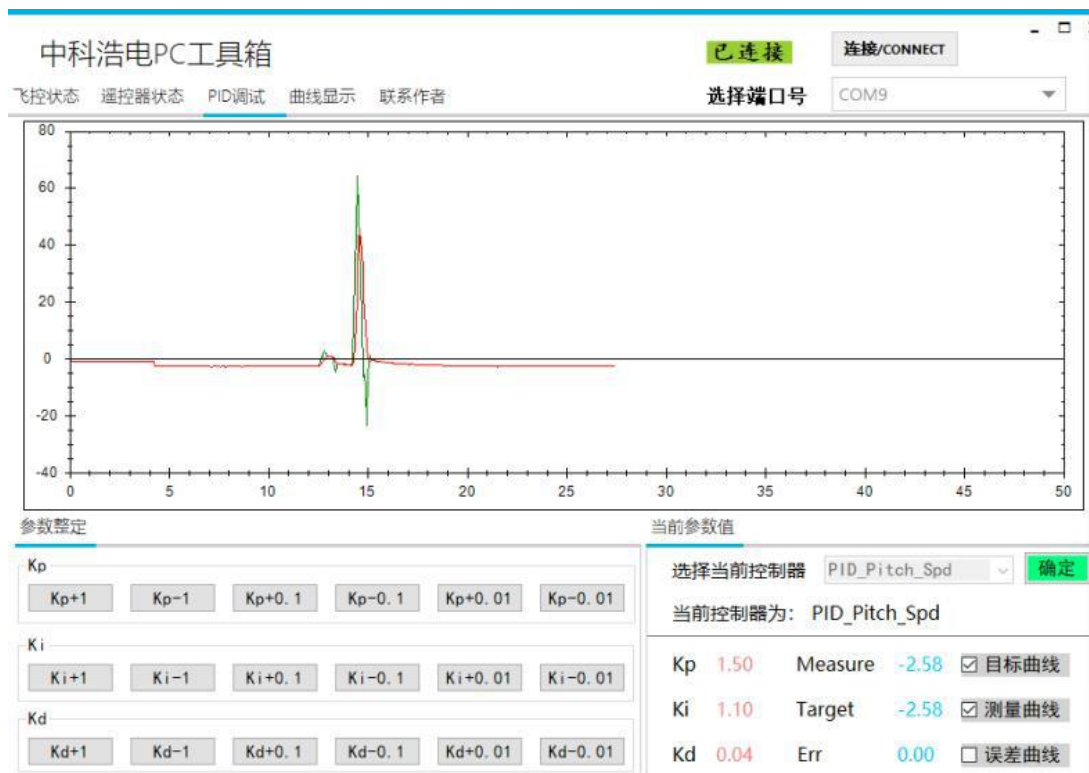


图 5-8 PID 数据测试图

5.5 飞行状态检测

无人机在飞行时一定要先检测飞行环境，无人机不是一个玩具，是螺旋桨一分钟转几千转的机器，因此需要有一个合适的环境进行飞行测试，避免无人机伤人事件的发生。无人通上电以后就可以打开飞控的开关了，接着会有一个校准的过程，判断无人机和遥控器是否能够进行正常的通信，如果是正常通信，飞控板上的蓝色 LED 灯会闪烁告知通信正常，通信失败则没有反应；检测到正常通信之后需要解锁飞控，飞机才可以正常飞行，解锁过程是遥控器的油门摇杆推到最上，接着将油门摇杆拉至最下，飞控核心解锁完成，这时缓缓推动油门后，即可开始飞行。当无人机达到一定的高度以后就可以开始进行前后左右的正常移动了，通过遥控器的方向摇杆控制无人机的飞行路径；若要使无人机平稳降落则需要缓慢将遥控器的油门摇杆推到最下，这样飞机就可以平稳降落了；如果飞行状态不稳定，无人机不受控制，则可以使用一键上锁的功能将飞机锁定，停止桨叶的转动，具体操作为将遥控器左右两边的摇杆同时向“内八”的方向拨动。如图 5-9 所示为无人机飞行状态图：

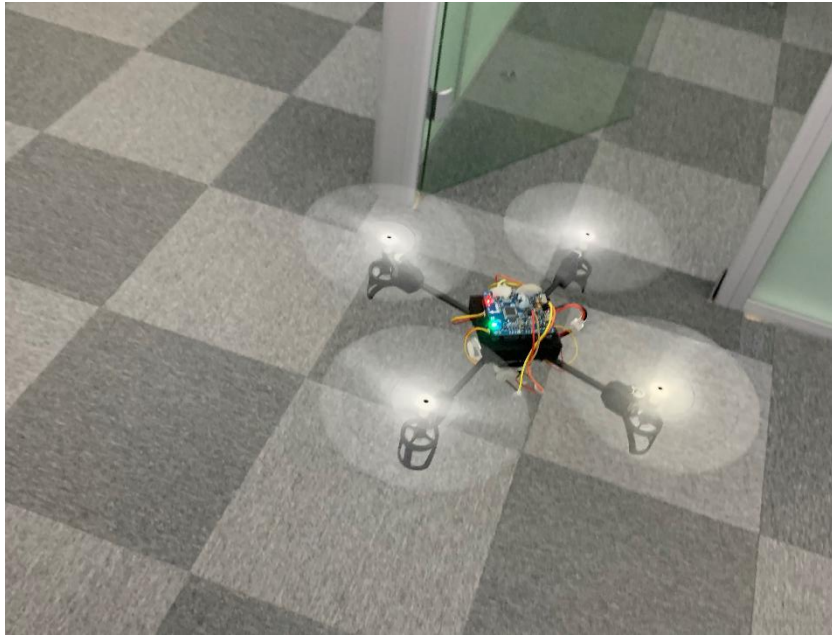


图 5-9 无人机飞行状态图

6 结束语

历时几个月的毕业设计终于做完了，由于本人技术有限，对于无人机相关知识的理解只是九牛一毛，因此只实现了几个最基本的功能，能够做到飞机正常起飞降落，在空中移动飞行轨迹，在紧急状态下的一键上锁，避免飞机伤人。在实际的开发过程中也遇到了很多问难，从对无人机的知之甚少，到做出一个系统的开发，每一步都会有各种各样的问题出现，从最开始的 PWM 信号驱动电机，不知道怎么实现驱动，到最后通过 PID 算法实现对无人机系统的控制，因为对 PID 算法从未了解，所以花费了很长时间才做好，在系统开发过程中也不断的请教别人和在网上查找资料，终于基本的功能都能实现，和预期效果相差无几，通过本次设计我也总结了系统的优缺点：

（1）系统的优点：

满足无人机正常飞行的需求，安全系数高，嵌入式知识应用非常多，跟随时代进步的脚步在新型领域探索。

（2）系统的缺点：

虽然基本功能都能实现，但飞机飞行状态不稳，容易蹿高，左右摇摆不定，系统的稳定性需要提高，而且无人机飞行对环境要求比较高。

参考文献

[1] 吴勇. 四轴飞行器 DIY 基于 STM32 微控制器[M]. 北京：北京航空航天大学出

- 版社, 2016:15-18.
- [2] 潘计辉, 张盛兵, 张小林. 飞行控制系统故障诊断与重构技术研究[J]. 计算机测量与控制, 2015, 4(4):1073-1074.
- [3] 董茂寅. 新课标互动同步训练物理教育科学版八年级下册[M]. 山东: 山东科学技术出版社, 2007:67-73.
- [4] 杜江冲. 基于北斗的无线地震勘探系统下位机研究[D]. 河北: 北华航天工业学院, 2018.
- [5] 张杰. 基于 MEMS 陀螺仪和加速度计的动态倾角传感器[J]. 机械设计与制造, 2012, 3(9):141—P143.
- [6] 陈星旭. 基于云计算的汽车运行状态监测与故障预警系统的硬件开发平台[D]. 重庆: 重庆交通大学, 2018.
- [7] 闫玺. 分数阶 PID 控制器及其参数整定的研究[D]. 西安: 西安理工大学, 2014
- [8] 胡庆. 基于 STM32 单片机的无人机飞行控制系统设计[D]. 南京: 南京航空航天大学, 2012.
- [9] 秦永元. 惯性导航[M]. 北京: 科学出版社, 2006: 288-312.
- [10] 白志刚. 自动调节系统解析与 PID 整定[M]. 北京: 化学工业出版社, 2012: 26-57.
- [11] Paul G.Fahlstrom, Thomas J.Gleason. 无人机系统导论[M]. 北京: 电子工业出版社, 2003: 22-48.
- [12] 尚和章. 小型无人机飞行控制系统硬件设计与实现[D]. 南京: 南京航空航天大学, 2012.
- [13] 梁延德, 程敏, 何福本, 李航. 基于互补滤波器的四旋翼飞行器姿态解算[J]. 传感器与微系统, 2011, 30(11):53-66.
- [14] 许中冲. 基于多传感器信息融合的无人机飞行控制系统设计[D]. 天津: 中国民航大学, 2016.

