东 莞 理 工 学 院

本 科 毕 业 设 计

|  |  |  |
| --- | --- | --- |
| **毕业设计题目：** | **便携式三轴手持云台的设计与实现** | |
| **学生姓名：** | **廖昌海** | |
| **学 号：** | **201341402502** | |
| **院 系：** | **计算机与网络安全学院** | |
| **专业班级：** | **2013级计算机科学与技术5班** | |
| **指导教师姓名及职称：** | | **黄培灿 实验师** |
| **起止时间： 2016 年 12 月 —— 2017 年 5 月** | | |

**摘要**

本课题设计了一种针对便携式摄影摄像设备的三轴手持云台。系统采用嵌入式微控制器STM32F103作为主控单元，6轴运动处理传感器MPU6050作为姿态反馈单元，三相无刷直流电机作为运动抵消单元，实现摄影摄像设备的稳定拍摄。为了验证该系统的可靠性，采取人体手持此云台进行跑步运动的方式，实验结果表明该云台已满足设计要求。

**关键词** 便携式、三轴、手持、云台

**Abstract**

The issue designed a three-axis hand-held pan-tilt for portable photographic cameras. The system uses the embedded microcontroller STM32F103 as the main control unit, the 6-axis motion processing sensor MPU6050 as the attitude feedback unit, the three-phase brushless DC motor as the motion cancellation unit, to achieve stable shooting of photographic cameras. In order to verify the reliability of the system, a human held the pan-tilt while ran, the experimental results show that the pan-tilt has fulfilled the design requirements.

**Keywords**  Portable, Three-axis, Hand-held, Pan-tilt

**目 录**

[第1章 绪论 1](#_Toc483135314)

[1.1 课题研究背景及意义 1](#_Toc483135315)

[1.2 相关课题的发展及现状 1](#_Toc483135316)

[第2章 系统结构和工作原理 3](#_Toc483135317)

[2.1 基本需求分析 3](#_Toc483135318)

[2.2 控制器系统结构 3](#_Toc483135319)

[2.3 三轴云台工作原理 4](#_Toc483135320)

[第3章 系统姿态解算和控制 6](#_Toc483135321)

[3.1 姿态解算 6](#_Toc483135322)

[3.1.1 惯性测量单元 6](#_Toc483135323)

[3.1.2 参考坐标系 8](#_Toc483135324)

[3.1.3 姿态解算算法 8](#_Toc483135325)

[3.2 姿态控制 10](#_Toc483135326)

[3.2.1 PID控制算法 10](#_Toc483135327)

[3.2.2 姿态控制量 11](#_Toc483135328)

[第4章 三相无刷直流电机控制 13](#_Toc483135329)

[4.1 无刷电机 13](#_Toc483135330)

[4.1.1 工作原理 13](#_Toc483135331)

[4.1.2 控制方法 15](#_Toc483135332)

[4.2 手控输入 16](#_Toc483135333)

[4.2.1 双轴按键摇杆模块 16](#_Toc483135334)

[4.2.2 基本功能与实现方式 16](#_Toc483135335)

[4.3 电源管理 17](#_Toc483135336)

[4.3.1 锂离子电池简介 17](#_Toc483135337)

[4.3.2 电池电压采集 17](#_Toc483135338)

[4.3.3 电池电压管理 18](#_Toc483135339)

[第5章 云台结构设计 19](#_Toc483135340)

[5.1 SolidWorks简介 19](#_Toc483135341)

[5.2 3D打印技术 19](#_Toc483135342)

[5.3 结构设计 20](#_Toc483135343)

[第6章 系统实现 23](#_Toc483135344)

[6.1 开发环境 23](#_Toc483135345)

[6.1.1 开发软件 23](#_Toc483135346)

[6.1.2 标准外设库 24](#_Toc483135347)

[6.1.3 工程配置 24](#_Toc483135348)

[6.2 编码实现 25](#_Toc483135349)

[6.2.1 系统基本功能 25](#_Toc483135350)

[6.2.2 系统电压与手控输入 27](#_Toc483135351)

[6.2.3 姿态更新与电机控制 28](#_Toc483135352)

[6.3 下载调试 29](#_Toc483135353)

[第7章 系统测试 30](#_Toc483135354)

[7.1 俯仰角和滚转角 30](#_Toc483135355)

[7.2 积分用Z轴角速度 30](#_Toc483135356)

[7.3 无刷电机SPWM波 32](#_Toc483135357)

[7.4 锂离子电池电压 33](#_Toc483135358)

[第8章 总结与展望 34](#_Toc483135359)

[8.1 全文总结 34](#_Toc483135360)

[8.2 课题展望 34](#_Toc483135361)

[参考文献 35](#_Toc483135362)

[致谢 36](#_Toc483135363)

# 绪论

## ****课题研究背景及意义****

近年来，随着科技的发展和生活水平的提高，手持通讯设备手机的性能逐步增强，普及率逐步提高，越来越广泛地被应用于摄影摄像、视频通话、视频直播等领域。在这些使用领域中，由于人体直接手持设备在运动过程中产生的振动，会使可移动设备对焦不准和画面模糊，是一项需要克服的难题。人们在生活中发现，鸟类的前庭眼反射机能能使视线内的目标处于稳定状态，俗称“鸡头稳定”，受此启发，自稳云台应运而生[1]。

目前市面上的自稳云台产品越来越多，但主要是针对专业用途，如摄像机稳定器、光电吊舱、四轴航拍器等。这些云台有的结构复杂难以便携使用，有的是价格昂贵难以普及应用。业界不乏对自稳云台的研究，但在便携式的小型云台的应用方面较少。这类小型云台可用于手机、运动相机、汽车行驶记录仪等，通过实时补偿拍摄设备的颠簸和抖动，在运动中捕捉清晰、稳定的影像，摆脱模糊的画面、歪斜的构图和杂乱无章的摇晃镜头。

基于此类实际应用需求的考虑，在研究的过程中采用模块化的设计思路，尽可能减小云台尺寸，简化云台结构设计，降低云台的成本，提升系统的稳定性。

## ****相关课题的发展及现状****

自稳云台的发展从纯机械式云台、半自动化的模拟云台到如今的全自动化数字电控云台，云台的控制系统综合了电机控制技术、传感器检测技术、微机电技术、捷联惯导技术和自动控制理论等多门学科。

自稳云台目前已主要应用与无人机上，广泛应用到航拍、勘测、现代农林业航空作业、电力和石油管道无人机巡线、自然灾害航空评估、自主侦察和电子对抗等各个方面，而无人机空中摄影则是应用最为广泛的一面。

大疆（DJI）当属无人机届翘楚，多年来凭借出色的航拍飞行器系列产品获得广大消费者的认可，其高品质的机载云台系统和航拍相机也备受瞩目。在2015 CES大展上，大疆首次将专业级航拍体验延伸到地面，正式推出灵眸Osmo手持云台相机，与众运动相机和三轴手持云台展开直接竞争。该机具备手持式三轴万向节稳定技术，俯仰、滚转、偏航轴可控角度分别为170°、100°、640°，最大控制转速为120°/s，内置980mAh容量电池，支持一个小时左右的拍摄，如图1-1所示。

图1-1 灵眸Osmo手持云台相机

Snappa Go是另一款手持云台稳定器，转为GoPro运动相机量身设计，是第一款具有集成控制功能，同时能控制GoPro相机和云台本身的手持云台稳定器,通过Snoppa自主开发的三轴增稳算法技术，可以时刻保持相机的平衡与流畅过渡，获得了2016近点设计奖年度最佳设计奖，如图1-2所示。

图1-2 Snappa Go手持云台稳定器

这些云台主要面对的是高端市场，对大众用户来说显得可望而不可及。从国内其他行业特点来看，大众需求是一个不可忽视的环节，是兵家必争之地，手持云台在此方面也同样具有广阔市场空间。一方面拍录早已成为刚需，另一方面大众用户需求庞大，这为手持云台实现大众化并做到全民普及打下先决基础与条件。未来两年手持云台将迎来爆发式增长。

# 系统结构和工作原理

## ****基本需求分析****

当下环境，自媒体等盛行，提高拍摄稳定性是一个较大的需求。三轴手持云台主要是在一般的运动环境下，在用户使用便携式摄影摄像设备（以下简称“相机”）进行拍摄时，消除人体给相机带来的振动，提高拍摄质量。

对于大众用户来说，用户收入适中，在设计云台时，应尽量降低云台成本，也应小型化、便携化。同时，云台系统的操作应简单明了，针对不同文化教育程度的用户，不需要专业指导便能上手操作。系统还应具有较强的鲁棒性，减少用户误操作造成的系统不稳定。

功能上，云台系统需要感知相机的姿态与变化趋势，并且能够迅速调整相机姿态，使相机保持稳定。为了防止电压过低影响云台性能等，需要对电源进行管理。功能实现均需要由一主控制器调度资源实现，主控制器要有较成熟的开发、调试解决方案。同时，加入手动控制模式能使用户使用方式更加多样化。

## ****控制器系统结构****

根据云台控制器的功能要求，本课题设计的云台控制系统主要由主控单元、姿态反馈单元、运动抵消单元、手控输入单元、电源管理单元五部分组成。主控单元是运算和控制中心，程序运行于此单元上；姿态反馈单元反映云台当前姿态与姿态变化趋势，为姿态调节奠定基础；运动抵消单元将根据主控单元计算出来的控制量，通过机械运动调节云台姿态；手控输入单元允许用户在程控和手控中切换控制方式，更能随心地控制云台运动；电源管理单元用于向系统供电并根据系统电压执行相关策略。主要元器件的选型如下：

1. 主控单元，选择了STM32F103RCT6作为主控器，封装如图2-1所示。此IC（Integrated Circuit，集成电路）使用了ARM Cortex-M3 32位的RISC（Reduced Instruction Set Computer，精简指令集计算机）内核，工作频率最高可达72MHz，内置256KB Flash和48KB SRAM，具有丰富的增强IO端口和联接到APB总线的外设，共有64个外部引脚，支持I2C、SPI、USART、CAN等标准通信接口。

图2-1 STM32F103RCT6

1. 姿态反馈单元，由两个6轴运动处理组MPU6050组成，MPU6050#1主要用于俯仰角Pitch和滚转角Roll的获取，#2主要用于偏航角Yaw的计算。MPU6050#1提供的姿态可直接用于俯仰和滚转电机的控制，#1和#2结合计算出来的角度差和#2的角速度用于偏航电机的控制。
2. 运动抵消单元，由三个三相无刷直流电机（Brushless Direct Current Motor，BLDCM）和三个半H桥驱动器DRV8313组成，DRV8313能将主控器的控制信号转换为三相变化的电压。
3. 手控输入单元，由一个双轴按键摇杆模块Joystick构成，Joystick同时带有按键和摇杆功能，可以用于模式切换、速度控制、角度控制等。
4. 电源管理单元，由一个11.1V 850mAh的锂离子电池LiPo、一个电源开关、两个LED指示灯和一个有源蜂鸣器Buzzer构成。绿色LED作为上电系统运行指示灯，红色LED和Buzzer用于LiPo低电压告警。

整个云台控制器的系统结构如图2-2所示。

图2-2 控制器系统结构

## ****三轴云台工作原理****

本课题设计的手持三轴自稳云台是一种搭载相机的具有俯仰、滚转、偏航3个自由度的装置，由空间上三个互相垂直的框架构成。其中，内框为俯仰框，中框为滚转框，外框为偏航框，分别负责控制相机的俯仰运动、滚转运动和偏航运动。相机固定于云台上，与云台顶部的姿态感应元件固联，通过捷联惯导技术，能直接敏感相机的线运动和角运动，解算出来的云台姿态变化即是相机姿态变化[2]。当云台载体受外界环境影响发生振动或倾斜时，安装在云台上的姿态感应元件就能够直接测量并获得相机视轴的偏差，经过控制算法计算并输出适当的控制信号来驱动相应的驱动电机，产生平衡力矩，使云台消除偏差，保持水平稳定[3]。由于云台系统处于高速计算当中，故当相机姿态稍微偏离正常姿态时，电机便快速反应修正姿态，保证了相机在拍摄过程中的稳定。但当俯仰轴、滚转轴和偏航轴中任一一轴的转动角度偏差大于可控角度时，即超出了云台能控范围时，该轴电机将停止运动。其中，可控角度受机械结构等因素影响，由程序进行角度限制。当角度偏差进入可控范围时，便可再继续控制。姿态控制流程图如图2-3所示。

图2-3 姿态控制流程图

因此，当人体手持云台在水平、垂直和旋转三个方向上运动时，会给相机带来相同的运动，云台系统通过机械运动来抵消人体带来的振动，使相机保持稳定，提高了相机的拍摄质量[4]。

# 系统姿态解算和控制

## ****姿态解算****

1. ****惯性测量单元****

姿态反馈单元是云台控制部分的核心，其中的IMU（Inertial Measurement Unit，惯性测量单元）一般由三轴加速度计、三轴陀螺仪和三轴磁强计组成。加速度计是测量自身加速度的装置。陀螺仪是一种用来传感与维持方向的装置，是基于角动量守恒理论设计出来的，主要是由一个位于轴心且可旋转的转子构成。由于转子的角动量，陀螺仪有抗拒方向改变的趋向[5]。三轴磁强计则可以根据测量的地磁场矢量计算确定地磁方向。考虑到磁强计易受电子设备干扰，故未采用。本课题采用的IMU为两个MPU6050。

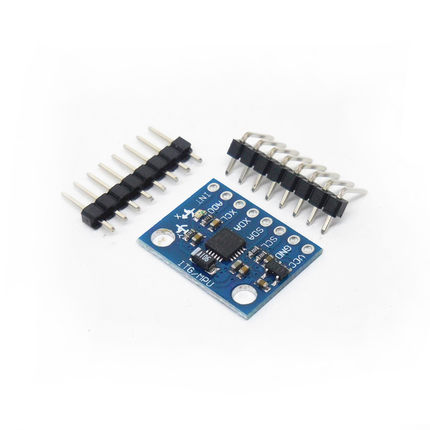
MPU6050是全球首例整合三轴MEMS（Micro-Electro-Mechanical System，微机电系统）加速度计和三轴MEMS陀螺仪的运动处理组件，免除了组合陀螺仪与加速度时间轴之差的问题，还有一个可扩展的数字运动处理器DMP，可用I2C接口连接一个第三方数字传感器，减少了大量的封装空间，传感器模块外型如图3-1所示，电路原理图如图3-2所示。

图3-1 惯性测量单元MPU6050

图3-2 MPU6050电路原理图

MPU6050对陀螺仪和加速度计分别用了三个16位ADC（Analog to Digital Converter，模数转换器）将其测量的模拟量转化为可输出的数字量。当传感器绕着任何一个轴旋转，通过电容器截止产生的振动引起科里奥利（Coriolis effect）效应，将得到的信号放大、解调和过滤，使产生的电压和角速度成正比，再通过ADC将电压转换成数字量[6]。

MPU6050拥有117个寄存器，可通过程序对寄存器进行配置，改变传感器的输出。地址0x1B的GYRO\_CONFIG寄存器，可编程配置陀螺仪的满量程范围，可配置的范围有±250、±500、±1000、±2000°/s（dps）。地址0x1C的ACCEL\_CONFIG寄存器，可编程配置加速度计满量程方位，可配置的范围有±2g、±4g、±8g、±16g。MPU6050内嵌的温度传感器可通过读取0x41和0x42获得一个16位数值，可用于温度补偿。

与MPU6050的通信是基于I2C串行总线。I2C是两线的总线，由串行数据SDA线和串行时钟SCL线在连接到总线的器件间传递信息。每个器件都一个唯一的地址识别，而且都可以作为一个发送器或接收器（由器件的功能决定）。同时I2C是多主机的总线，数据传输速率在标准模式下可达100kbit/s，在快速模式下可达400Kbit/s，在高速模式下可达3.4Mbit/s。

STM32的硬件I2C尚且存在缺陷，不稳定，可能导致死机，故使用STM32的GPIO引脚模拟I2C。模拟I2C即是将引脚设为开漏输出，然后手动改变引脚电平来仿真I2C通信协议。

传输起始条件由唯一的主机产生，表示与从机的数据传输的开始。当SCL线保持高电平时，SDA线电平从高电平转换为低电平，即产生了起始信号。

传输停止条件也由主机产生，表示与从机的数据传输的结束。当SCL线保持高电平时，SDA线电平从低电平转换为高电平，即产生了停止信号。

通过I2C总线发送的每个字节（8 bits）之后要跟一个接收器的确认条件，即主机在拉低SCL线以完成一个字节的第8位传输之后，接收器在第9位的时间内将SDA线拉低或者拉高。拉低表示接收器正确接收到数据，为ACK；拉高则SDA线保持高电平，表示接收器未正确接收到数据，为NACK。如果缺少ACK信号，则从设备会中止传输进入空闲模式。

一次完整的I2C数据传输为，主机发送起始信号，发送从设备地址，地址为8位，前7位为器件地址，后1位表示，地址匹配的从设备发送ACK信号便与主机开始数据传输，每传输一个字节接收器件发送ACK，由主机发送停止信号或者接收设备发送NACK停止传输。完整的数据传输如图3-3所示。

图3-3 I2C完整的数据传输

1. ****参考坐标系****

本课题设计的三轴云台在空间上是由三个互相垂直的俯仰轴、滚转轴和偏航轴组成，使用的是载体坐标系（b系），它是一个相互正交的坐标系。由于IMU是安装在云台上与云台固联，因此测量的传感器值是在载体坐标系中的值。将云台上的俯仰轴、滚转轴和偏航轴在载体坐标系上分别定义为X、Y、Z轴，则X轴沿载体（云台）横轴指右，Y轴沿载体纵轴指前，Z轴沿载体竖轴并与X、Y轴构成右手直角坐标系，如图3-4所示，α、β、γ分别为俯仰角、滚转角和偏航角。

图3-4 载体坐标系

1. ****姿态解算算法****

**姿态解算是将陀螺仪和加速度计的数据融合在一起，通过滤波处理得出的旋转姿态的过程。当MPU6050正面朝上放置与水平面上时，X轴和Y轴的加速度大小为0g，Z轴加速度大小为1g，由牛顿第三定律可知，加速度方向沿Z轴正方向。当控制器初始化设置完毕后，相机姿态发生变化时，MPU6050返回的三轴加速度受重力分量影响亦发生相应变化。因此，可以根据三轴分量关系计算出姿态角。**

**俯仰角计算示意图如图3-5所示。**

**图3-5 俯仰角计算示意图**

**由图可知，俯仰角的计算公式为：**

**(3.1)**

**其中，Ay、Az分别表示Y轴和Z轴加速度值。**

**滚转角计算示意图如图3-6所示。**

**图**3-6 滚转角计算示意图

**由图可知，滚转角的计算公式为：**

**(3.2)**

**其中，Ax、Az分别表示X轴和Z轴加速度值。**

**偏航角无法通过三轴加速度计算出来。同时，加速度计需要依靠系统外绝对参照物重力，在无外力加速度的情况下，能准确输出俯仰角和滚转角，并且此角度不会有累积误差，在更长的时间尺度内都是准确的。但是加速度计实际上是用MEMS技术检测惯性力造成的微小形变，而惯性力和重力本质上是相同的，故它无法分辨重力加速度和外力加速度。当系统在三维空间做变速运动时，它的输出就不准确了，就比如本课题的使用环境。同时，也存在动态响应慢的问题。因此，需要陀螺仪提供的三轴角速度来计算姿态角。**

**三轴的角度都可通过对角速度的积分来计算，通用的计算公式如下：**

**(3.3)**

**其中，anglen为第n个周期角度值；anglen-1为第n-1个周期角度值；angular\_V为对应轴的角速率；Tcal为积分周期。**

**对于单轴陀螺仪来说，影响其静态测量精度的主要因素是该传感器的零偏误差、刻度系数误差和随机漂移误差；但对于三轴陀螺仪来说，其测量结果的精度与构成三轴陀螺仪的各单轴陀螺仪的零偏误差、刻度系数误差、随机漂移误差以及各单轴陀螺仪敏感轴之间的不正交安装误差相关[7]。积分过程是一个时间推移的过程，这个过程中的瞬时精度高，但是随着时间的累积相对误差逐渐变大，长时精度变低。所以需要本课题使用一阶互补滤波算法对加速度和角速度分别计算的姿态角进行数据融合，以获得更加精确的加速度值。**

**(3.4)**

**其中，anglen为第n个周期角度值；calWeight为加速度计数据计算所得的角度的计算权重；angle\_A为加速度计数据计算所得的角度；angle\_G为陀螺仪数据计算所得的角度。**

## ****姿态控制****

1. ****PID控制算法****

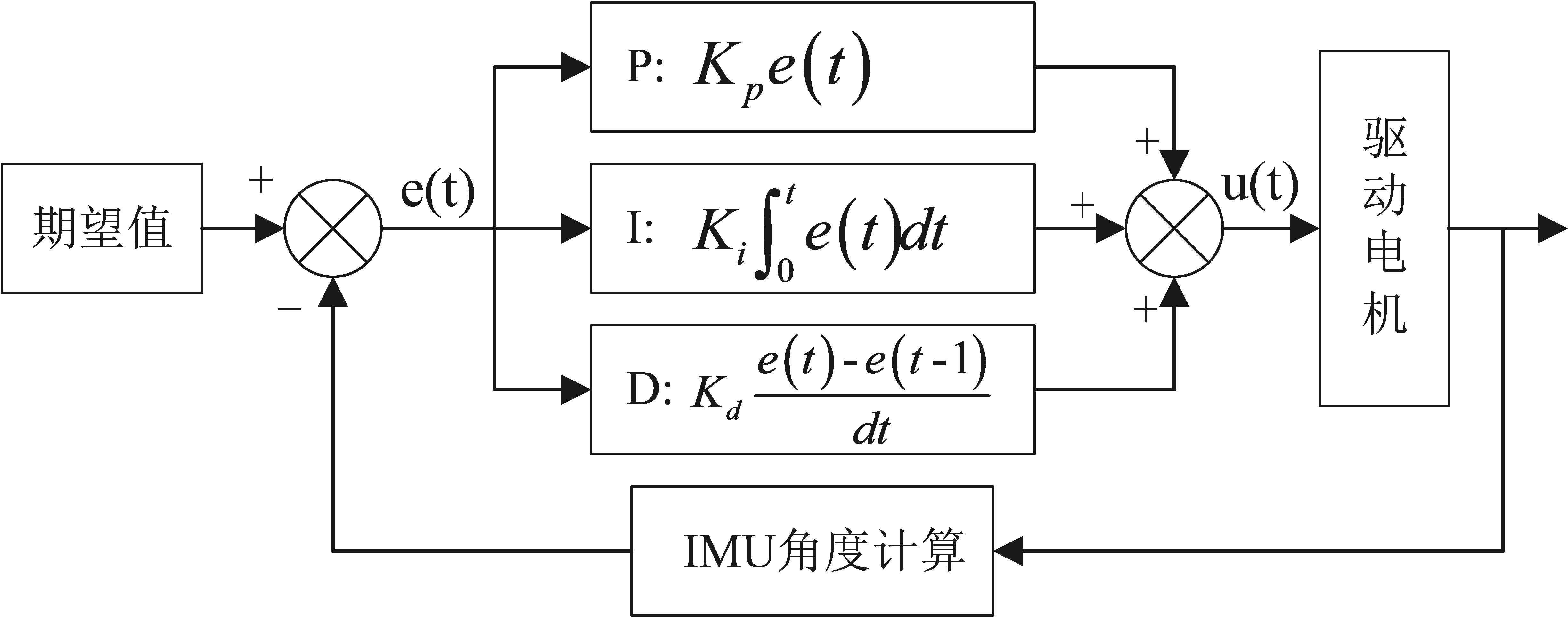
由IMU和姿态解算算法，可以求出姿态角，并作为后续PID控制器的输入。本课题采用的单环位置式PID控制器，通过角度反馈和期望角度来计算输出电机转动方向及速度，即通过控制云台的角度来达到控制云台的稳定。PID控制框图如图3-7所示。

图3-7 PID控制框图

PID控制器的计算公式为：

(3.5)

其中，u(t)为输出给电机的控制量；Kp为比例系数；e(t)为实际值与期望值的偏差；Ki为积分系数；积分内容为从0时刻到t时刻的累积误差；Kd为微分系数；微分内容为偏差的变化率。

比例环节是PID控制器中必不可少的环节。比例环节的作用为放大误差信号，提高控制器对于偏差信号的感应灵敏度，其特点是不失真、不延时、成比例地复现控制器输入信号的变化。较大的比例系数能加快调节误差，但过大的比例系数会使系统的稳定性降低，增加超调量，出现振荡甚至发散。比例系数的选择只能在稳定性和灵敏性之间进行折中选择。

积分环节通过将实际值与期望值的偏差不断进行累积，是控制器的输出信号不断增强，直到偏差为0，从而消除系统的静态误差。较小的积分系数可以减小超调，提高动态响应的平稳性，但消除静态误差的时间较长。过大的积分系数可能引起系统振荡和积分饱和，将带来较大的超调量并延缓进入稳态的速度。

微分环节根据偏差的变化趋势输出控制量，并能在偏差值发生较大变化之前输出超前校正信号。微分环节可以使系统的超调量下降，同时改善系统的动态调节速度。过大的微分系数会使系统响应过程提前制动，从而延长调节时间并出现余差，同时会削弱系统的抗干扰能力。但值越大，稳定系统的静态误差就会消除的越快[8]。

1. ****姿态控制量****

姿态解算出来的俯仰角和滚转角可以直接输入PID控制器，得到相应的控制量给电机。而偏航角需要进行额外的处理。由于无法通过加速度计计算出偏航角角度，所以只能通过对陀螺仪的Z轴角速度进行积分得到。但由于陀螺仪存在的较大偏差问题，便在初始化时需要对对陀螺仪Z轴角速度进行偏差校准，并根据计算得到的角速度的大小进行不同的控制处理。

偏航电机的基本控制策略为，当位于手持手柄上的MPU6050#2的Z轴角速度大于设定阈值时，开环控制电机，将角速度比例放大后作为电机运动速度，电机运动方向与手柄旋转方向相反；当Z轴角速度小于设定阈值时，将MPU6050#2积分出来的偏航角与位于相机上方的MPU6050#1积分出来的角度相减，将差值作为PID控制器的输入，以实现相机偏航运动的平稳过渡。

开环控制中，角速度的数据变化快，波动大，无法有效地表示一个运动过程，故需要对Z轴角速度进行滤波。本课题采用的是中值滤波法。

中值滤波法要求有一段固定的空间来缓存数据，每个周期将一个最旧的值替换为最新的值，再进行排序，取出位于中间的值。本课题实际使用时进行了稍微变形，取出的是中间值左右各N/2个，占总缓存数据量M%的数据的均值。N由缓存长度和M确定，M%小于1。中值滤波的基本流程图如图3-8所示。

图3-8 中值滤波基本流程图

中值滤波能消除Z轴角速度一定的噪声，但还存在这一个由陀螺仪传感器本身的特性引起的偏差，其中包括零点漂移。所以，不管是开环还是闭环控制，对Z轴的角速度进行其他处理之前，需要先做一个偏差校准，主要是消除Z轴角速度的零点漂移问题。

系统上电初始化时，云台会转动至水平面，然后静置一段时间进行Z轴角速度累计求和，并在之后求取均值，将均值作为新的零点，如下式所示：

(3.6)

其中，Zero为新的零点；times表示累加次数；Gz为陀螺仪Z轴角速度。

消除零点漂移后，便可对Z轴角速度进行中值滤波或者积分操作。

# 三相无刷直流电机控制

## ****无刷电机****

1. ****工作原理****

无刷直流电机是发展快速且应用广泛的电机种类，它使用半导体逆变器的电子换向取代一般直流电机中的机械换向，本质上属于永磁同步电机。与有刷电机相比，无刷电机能获得更好的扭矩转速特性、高速动态响应和高效率长寿命等。但特性上若要达到相当于直流电机的性能须使用复杂控制技术才能达到。现今半导体发展迅速，功率组件的切换频率加快许多，提升了驱动电机的能力。无刷电机更优的扭矩和外形尺寸比使得它更适合用于对电机自身重量和大小比较敏感的场合。

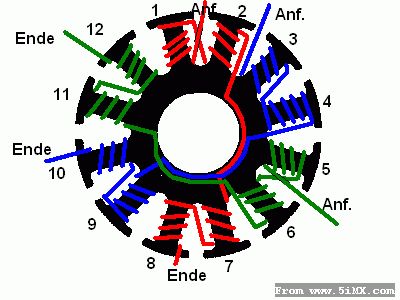
无刷电机主体一般由定子和转子组成。定子是由许多硅钢片经过叠压和轴向冲压而成，每个冲槽内都有一定的漆包线组成绕组，通常匝数越多扭矩越大。大多数定子的绕组呈星型排列，每个绕组按照一定的顺序在槽上缠绕。内转子型电机的转子是由磁钢（永磁体）按照N极和S极交替排列在转子周围构成的，外转子型电机转子的磁钢则是贴在转子内壁。一般来说，磁钢数越多，扭矩越大。业界 规定，槽数必须是3的倍数，磁钢数必须是偶数，且槽数和磁钢数之间不能整除。一种12N（槽）14P（极）电机绕组示意图如图4-1所示。两种12N14P外转子型电机内部结构图如图4-2所示。

图4-1 一种12N14P电机绕组示意图

图4-2 两种12N14P外转子型电机内部结构图

本课题使用的无刷电机为GB2208，如图4-3所示，属于外转子型电机。电机槽极为12N14P，齿槽转矩小，绕组系数较高，扭矩在12N的电机里较大，扭力大小为每3S（11.1V）为0.6kg·cm，重量为42g，尺寸图如图4-4所示。

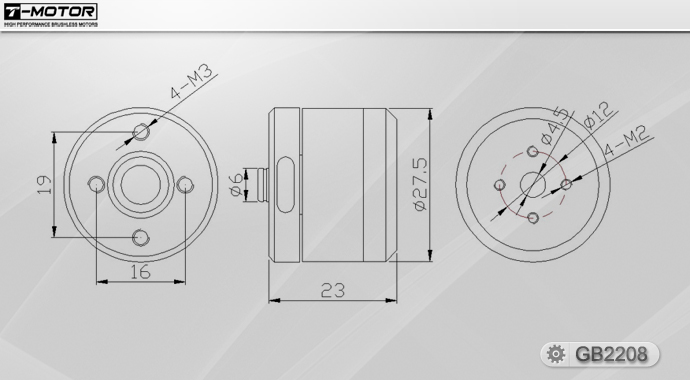
图4-3 三相无刷直流电机GB2208

图4-4 无刷电机GB2208尺寸图

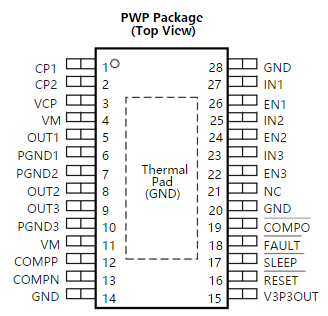
要让电机转动起来，就需要在每一次换向的时候，使一组绕组处于正向通电状态，第二组处于反向通电状态，第三组不通电。这样，转子产生的磁场和定子磁钢的磁场相互作用产生了转矩。理论上，当这两个磁场夹角为90°时转矩最大，当这两个磁场重合时转矩为0。为了使转子不停转动，就需要不停按顺序改变定子的磁场，像转子的磁场不断追赶定子的磁场一样。典型的三相六状态电机运动示意图如图4-5所示，电机绕组通电时序图如图4-6所示。

图4-5 三相六状态电机运动示意图

图4-6 三相六状态电机绕组通电时序图

1. ****控制方法****

**为了给电机输出一个规律变化的通电状态，可以通过三组互补的PWM（PulseWidth Modulation，脉冲宽度调制）来导通与关断六个开关功率管实现，但需要占用主控器太多定时器且成本高、功耗大，故使用常用来驱动三相无刷直流电机的DRV8313。DRV8313封装示意图如图4-7所示，引脚示意图如图4-8所示。**

**图4-7 DRV8313封装示意图**

**图4-8 DRV8313引脚示意图**

**DRV8313提供了三个独立控制的半H桥驱动器，当INx变化时，OUTx的开关延时时间不超过500ns。实际使用时，ENx始终处于有效状态，通过对INx输入正弦调制的PWM波，即SPWM（Sinusoidal PWM）波，OUTx输出给电机的电压呈正弦变化。通过SPWM波控制逆变电路中开关器件的通断，使其输出的脉冲电压的面积与所希望输出的正弦波在相应区间内的面积相等，通过改变SPWM波的频率和幅值就可调节逆变电路输出的电压的频率和幅值。**

**在程序中将正弦波划分为N等份，每一份对应的正弦值即是PWM波的占空比，使用一个静态数组存储起来。三相无刷电机需要三个输入，每相输入的SPWM波的相位差为N/3，经过M个周期T，三相输入的PWM波的占空比一同改变为数组的下（顺时针旋转）或前（逆时针旋转）一索引的值。通过改变N的值可以改变对电机的控制精度；通过改变M和T的值可以改变电机转动的速度。**

## ****手控输入****

1. ****双轴按键摇杆模块****

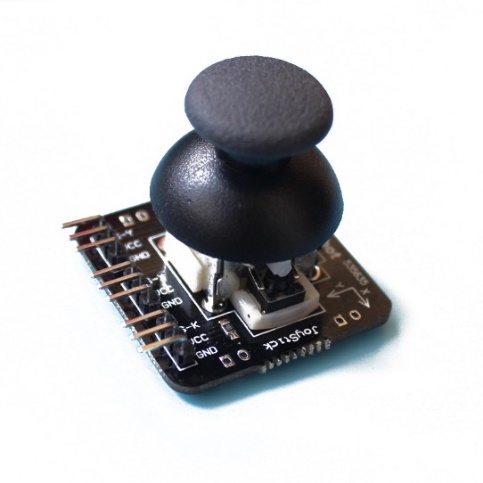
系统的手控输入使用的是双轴按键摇杆传感器模块，采用PS2摇杆电位器制作，具有X、Y两轴两路模拟输出，Z轴一路按钮数字开关量输出，X、Y、Z轴电路都单独引出，如图4-9所示。

图4-9 双轴按键摇杆模块

两路模拟输出分别对应X、Y轴偏移量，开关量则表示用户是否在Z轴按下。模块输入的电压范围为直流3.3V至5.0V，模拟量的输出为0V至对应使用的供电电压，数字量的输出为相应的高低电平。

模块工作原理较为简单，十字摇杆本质上为两个双向的10K变阻器，随着摇杆的方向不同阻值也不同，输出的电压也相应地发生变化。主控器直接连接模块，使用AD采集便能获取摇杆状态，进行相应控制。

1. ****基本功能与实现方式****

**Z轴按键用于系统自控模式和手控模式的切换。主控器不断扫描引脚电平，当按键按下时引起电平变化，在程序中便能实现控制模式的翻转。**

**X、Y轴摇杆用于分别控制云台的竖直和水平运动，相对应的分别是俯仰角和偏航角。通过AD采集获取摇杆状态，通过摇杆偏离中立位的方向和差值，来赋予电机相应的运动方向与速度。同样的，AD采集到的摇杆数据也需要先经过滤波处理方可使用。**

## ****电源管理****

1. ****锂离子电池简介****

云台系统使用锂离子电池作为系统电源。锂离子电池是由日本索尼公司发明的一类以碳材料为负极，以含锂的化合物作为正极的电池，通常情况下指锂二次电池（充电电池）。在充电过程中，含锂化合物的锂离子脱出，经过电解液运动到负极，嵌入到负极层状的炭材料中。放电过程锂离子的运动方向相反。

锂离子的能量密度大，平均输出电压高，自放电率小，每月在10%以下，并且没有记忆效应，循环性能优越，可快速充放电。单个电芯的锂离子电池的额定电压一般为3.7V，充电截止电压不得超过4.2V，放电截止电压以3.5V为宜。本课题采用的是3S 850mAh的锂离子电池,如图4-10所示。

图4-10 3S 850mAh锂离子电池

3S表示该电池包为3个电芯串联，整体输出电压为11.1V，充电截止电压为12.6V，放电截止电压为10.5V。850mAh表示电池的整体容量。30C指的是电池的充放电倍率，指电池在规定时间内放出其额定容量时所需要的电流值，由于电池的标称额定容量为850mAh，故放电电流最大为25.5A。

1. ****电池电压采集****

电池电压采集使用的是主控器的ADC采集。STM32拥有3个12位的模数转换器，1us的转换时间，转换范围为0~3.3V。系统使用的11.1V电池远远超过ADC采集范围，故需要进行分压后再输入ADC，分压采集电路如图4-11所示。

图4-11 LiPo分压采集电路

对于ADC采集到的模拟量计算，有两种方法。第一种是使用稳压源或者调压模块作为输入，根据采集到的模拟量与仿真输入的已知电压进行数据拟合，模拟量作为自变量，已知电压作为因变量，可以拟合得到一一元一次或二次方程，多次方程可能影响微处理器性能。第二种是根据ADC特性先计算出采集到的电压值，然后根据分压采集电路逆推出LiPo电压。

本课题采用的是第二种方法，计算公式如下：

(4.1)

其中，Vadc为ADC采集得到的电压值；analog为ADC采集到的模拟量；Vlipo为LiPo当前电压值。得到结果后对结果进行一阶数字低通滤波处理。

(4.2)

其中，Yn为本次滤波的输出值；a为滤波系数，即本次采样值对滤波输出的贡献；Xn为当前采样值；Yn-1为上次滤波输出值。

1. ****电池电压管理****

锂离子电池放完内部存储的电量，电压到达一定值，一般是3.5V左右，继续放电就会造成过度放电，对电池造成永久性损坏。一般来说，过度放电会使电池内压升高，内阻增加，正负活性物质可逆性受到破坏，电解液分解，负极层状结构出现塌陷，限制充电过程中嵌入的锂离子数量。即使充电也只能部分恢复，容量也会有明显衰退，寿命缩短且不可以恢复。同时，系统电压低于一定阈值时，会造成无刷电机扭矩不足，云台系统将无法正常工作。

因此，有必要对锂离子电池进行电压管理。当电池电压低于设定阈值T时，主控器控制红色LED频闪，蜂鸣器频响，用于提醒用户。同时，程序自动关闭SPWM波输出，则电机驱动器停止输出电压给电机，电机处于未上电状态。此时，需要关闭电源更换电池。即使电压采集发生跳变，电压突变为正常工作电压，系统也将保持告警状态。即，一旦进入电压告警状态，则只能断电重启系统。

阈值T需要根据所使用的锂离子电池的组建方式确定。若锂离子电池的电芯接法为x S（串联）y P（并联），且放电截止电压以3.5V为准，则：

(4.3)

# 云台结构设计

## ****SolidWorks简介****

SolidWorks是一个在Windows环境下进行机械设计的软件，是一个以设计功能为主的CAD软件，其界面完全使用Windows风格，具有人性化的操作界面。Solidworks在用户界面方面的方便程度是公认的。它能完全自动捕捉用户设计意图和引导用户设计修改，并且可以动态修改特征和草图，实时预览效果。在SolidWorks的装配环境里，可以利用已有的三维零件模型，按照一定的约束关系进行组装，形成产品的虚拟装配，还可以进行运动分析等，以验证零件的尺寸等。利用零件及其装配的实体模型，可以自动生成零件及装配的工程图。指定模型的投影方向或者剖切位置等，就可以得到需要的图形。同时，工程图是相关的，当修改图纸的尺寸时，零件模型、视图、装配体都将自动更新。SolidWorks数据接口支持的标准多，转换成功率高，这也为3D打印奠定基础。因为完成结构设计时，需要将文件转换为3D打机识别的文件类型\*.STL。

SolidWorks十分容易掌握和使用，它的三大特点便是功能强大、易学易用和技术创新，这也使得Solidworks成为领先的、主流的CAD解决方案。

## ****3D打印技术****

3D打印（3D Printing，又称三维打印）是一种快速成形技术，它以数字化模型为基础，运用粉末状金属或塑料等可粘合材料，通过逐层打印的方式构造物体。 由于其在制造工艺方面的创新，被认为是“第三次工业革命的重要生产工具”。3D打印技术早在 20 世纪 90 年代中期就已出现，但由于价格昂贵，技术不成熟，早期并没有得到推广普及[9]。但相比于传统的制造技术，3D打印制造门槛较低，即使是复杂的物体，由于不必事先制造模具，也不必通过复杂的加工锻造工艺，只需要在计算机中利用建模软件制作模型然后打印即可获得最终产品，降低了制造的难度，同时在生产上可以实现结构优化、节约材料和节省能源。

一般3D打印流程为，三维建模、打印配置、打印、后期处理。

三维建模可以通过3D扫描仪获取对象三维数据，并且以数字化方式生成。也可以使用Solidworks等三维建模软件从零开始建立三维数字化模型，或是直接使用他人已做好的模型。然后将模型数据转换为3D打印机能识别的类型，通过打印机配备的专业软件对打印方式进行进一步处理并发送给打印机。打印是打印机将打印耗材逐层叠加到三维空间中，根据打印材质不同，打印所使用的设备不同，工作原理也不同。整个过程根据模型大小、复杂程度、打印材质等耗时数分钟到数天不等。模型打印完成后一般都会有毛刺或是粗糙的截面，需要对模型进行后期的加工处理。

3D打印技术用途广泛。军事上快速制造舰船配件；航空上打印无人飞行器；医学上打印人体骨骼、器官等；还可以打印房屋建筑、汽车、电子设备、服装服饰等。3D打印技术正在逐步改善以至能够普遍推广使用，我国也正加快发展先进制造和3D打印，紧跟世界脚步。

## ****结构设计****

本课题3D打印的结构均使用PLA（Plylactic Acid，聚乳酸）作为打印耗材。PLA是一种新型的生物基及可生物降解材料，使用可再生的植物资源（如玉米）所提出的淀粉原料制成，热稳定性好，加工温度170～230℃，有好的抗溶剂性，可用多种方式进行加工。PLA几乎不会收缩，即使是开放式的打印机，也能打印巨大的物体，也不必担心成品从板子上悬空、歪斜或破损，也适合实地在公共场所做3D打印。但PLA打印的成品不适合在60℃以上的环境温度里使用，这样的温度会使材料受热形变。由于在使用时电机转速较低，热量无法散发出去，第2版设计的结构在测试中发现了结构形变问题，所以在绘制第3版结构时，使用了航模中22系列电机通用的铝合金散热式电机座，如图5-1所示。设计结构时，需要先测量电机座的尺寸，再结合在航模中的使用方式设计连接部分的模型。成品及安装方式如图5-2所示。

图5-1 铝合金电机座

图5-2 电机座安装示意图

虽然PLA也能打印出强度相当高的物体，却比其他塑料稍微脆弱一点。它不具有弹性，掉落或撞到东西时，多半会产生缺口或破损，而不会弹回来。只要稍微弯曲就会折断，不适合做成薄的东西。第2版结构以前，均使用填充密度低于50%的打印方案，即结构内部填充的百分比低于50%，这就造成了本来强度不大的PLA更加易于损坏，最明显的是相机固定结构的损坏。故从第2版结构开始，均使用100%的填充密度，并且在多处增加圆角以增加结构强度。

俯仰结构中，最重要的是相机固定结构。这里使用的使用的是夹块结构，有两种方案，第一种是像如今许多手持自拍杆一样，通过弹簧的弹力作用固定相机；第二种是通过螺杆的旋进旋出来根据相机大小调整夹合大小。考虑到弹簧弹力难以计算，弹力过小难以夹住相机，弹力过大用户难以伸展开结构，故使用第二种方法，结构设计和安装示意图如图5-3所示。

图5-3 相机固定结构设计与安装示意图

**横滚和偏航结构使用了椭圆曲线设计，在设计时还考虑了实际安装时需要使用工具，各保留了一个安装孔，用于通过螺丝刀。结构示意图如5-4所示。**

**图5-4 横滚与偏航结构示意图**

使用3D打印机时，需要考虑打印物体的大小及打印过程中的受力变化，防止打印过程中受力变化导致物体倾倒等。在打印第2版手柄结构时，由于头重脚轻，底部无支撑结构，导致在打印过程中发生倾倒，打印失败，造成打印材料和时间的浪费。故在第3版结构中，将手柄结构分段拆分，使用转动连接，为了连接稳定性，增加了M3螺孔用于固定。同时，为了保持打印过程中的强度和稳定，参照了树状、桥形等支撑结构[10]，在模型底部增加了相应支撑。手柄结构如图5-5所示。

图5-5 手柄结构示意图

其中，第一部分结构用于连接金属电机座与固定控制板；第二部分结构，前面板L型结构用于固定双轴按键摇杆模块，右侧矩形槽用于嵌入电源开关，后半部分挖空的空间用于放置锂离子电池和多余的线缆；第三部分结构仅为手持时握住的手柄；小圆柱形支撑结构将在打印完成后进行后期处理时去掉。

# 系统实现

## ****开发环境****

1. 开发软件

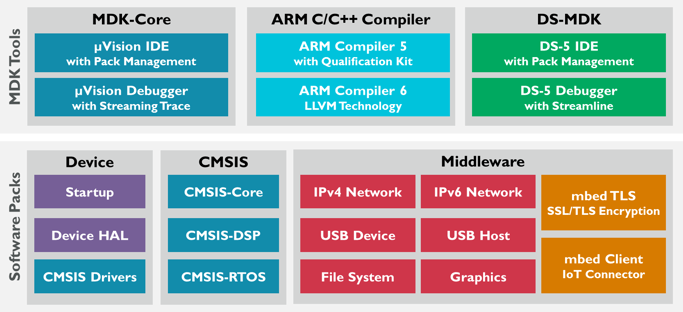
Keil MDK（Microcontroller Development Kit，微控制器开发套件）是美国Keil Software公司开发的针对基于ARM的微控制器最全面的软件开发解决方案，并包含创建，构建和调试嵌入式应用程序所需的所有组件。MDK包括μVisionIDE和调试器，ARM C / C ++编译器和必不可少的中间件组件，如图6-1所示。它专为微控制器应用而设计，支持超过3900个设备，支持第三方调试适配器和ULINK系列调试与跟踪适配器，不仅易于学习和使用，而且功能强大，能够满足大多数苛刻的嵌入式应用。

图6-1 MDK产品组件

MDK-ARM有四个可用版本，分别是MDK-Lite、MDK-Basic、MDK-Standard、MDK-Professional。所有版本均提供一个完善的C / C++开发环境，其中MDK-Professional还包含大量的中间库。MDK-Lite针对的是产品评估、小项目和教育，可任意下载使用，不需要序列号或许可密钥，代码大小限制在32KB。MDK-Lite通过使用一个有效的产品序列号可以转换成其他MDK-ARM版本。

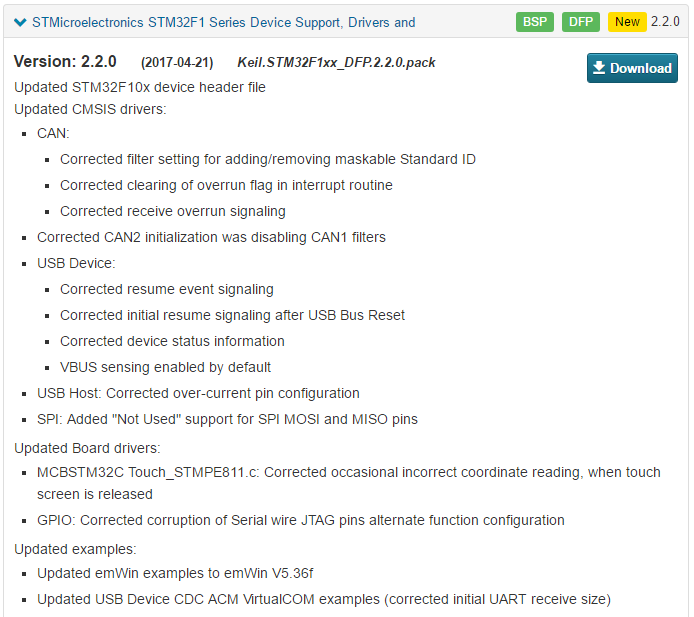
本课题使用的是MDK-Lite版本，安装完软件后，还需要根据使用的微控制器添加设备支持包。由于云台系统的主控器的是STM32F103RCT6，所以选择STM32F1系列的设备支持包，可以在MDK官网下载，如图6-2所示。

图6-2 STM32F1系列设备支持包

MDK专注于支持嵌入式设备的开发调试上，在代码编辑方面不够智能，故扮演的是编译和调试工程代码的角色，扮演编辑器角色的是Source Insight。

Source Insight是一个面向项目开发的程序编辑器和代码浏览器，它拥有内置的对C/C++等程序的动态分析，能快速解析整个项目，自动创建高性能的符号数据库，包括函数、全局变量、结构体等和工程源文件里定义的其它类型的符号，并在工作的同时动态维护数据库，自动显示有用的上下文信息，编辑时自动完成代码片段、符号。同时，Source Insight通过类树图显示函数与函数、函数与变量间的调用关系，能快速查找函数、变量的引用。

1. 标准外设库

传统的单片机开发，通过直接操纵单片机寄存器，进而控制芯片的工作方式，如中断、定时器等。但随着单片机片上外设资源越来越丰富，寄存器数量和配置复杂度增加，造成开发速度慢、程序可读性差等，直接影响了开发效率和程序维护成本，而库开发方式正好弥补了这些缺陷。

STM32库是由ST（意法半导体）公司针对STM32提供的函数接口，该函数库是一个固件函数包，由程序、数据结构和宏组成，包括了微控制器所有外设的性能特征。该函数库还包括每一个外设驱动描述和应用实例。通过该函数库，无需深入掌握细节，用户也可以轻松应对每一个外设。因此，使用函数库可以大大减少用户的程序编写时间，进而降低开发成本。

每个外设驱动都由一组函数组成，这组函数覆盖了该外设所有功能。每个器件的开发都由一个通用API（Application Programming Interface，应用编程界面）驱动，API对该驱动程序的结构、函数、参数名称都进行了标准化。

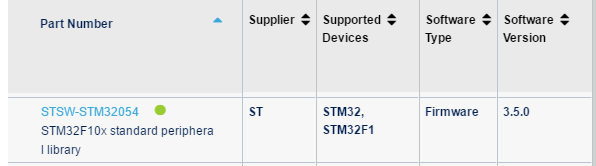
本课题使用的STM32F10x标准外设库，针对的是STM32F10x系列芯片，可以在ST官网下载，如图6-3所示。

图6-3 STM32F10x标准外设库

1. 工程配置

使用MDK新建一个工程，选择CPU为ST公司的STM32F103RC。由于使用了STM32标准外设库，所以在新建工程后，需要将启动文件和在开发过程中使用到的相关的库文件添加到工程中。同时，在目标选项中的预处理符号中添加定义STM32F10X\_HD和USE\_STDPERIPH\_DRIVER。STM32F10X\_HD表示高密度的STM32设备，即Flash大小在256KB~512KB的设备；USE\_STDPERIPH\_DRIVER表示使用STM32标准外设库，类似一个开关作用，如若无定义，则外设库的头文件均不会被包含且编译。这两个宏定义也可以在stm32f10x.h文件中找到官方的相关说明。在包含路径里，需要把库函数的相关头文件的路径添加进来，在实现系统过程中，还需要不断添加用户编写的头文件路径。基本配置步骤如图6-4所示。

图6-4 基本配置步骤

在调试选项卡中，由于没有使用外部调试器，所以选择仿真器，并修改CPU DLL文本框中文本为SARMCM3.DLL；Dialog DLL文本框中的为DARMSTM.DLL，其参数为-pSTM32F103RC。

## ****编码实现****

1. 系统基本功能

为了判断系统是否上电或者系统程序是否正常运行，需要具有指示意义的LED，所以LED状态在程序运行时需要进行开关和翻转操作，在编写此类操作时，使用了函数宏直接修改寄存器的方式，具有一定的可扩展性。如下为两个LED的操作代码的宏实现。

#define BIT\_SET(port, pin) (port->BSRR = pin)

#define BIT\_RESET(port, pin) (port->BRR = pin)

#define BIT\_REVERSE(port, pin) (port->ODR ^= pin)

#define LED0\_ON BIT\_SET(GPIOB, GPIO\_Pin\_12)

#define LED0\_OFF BIT\_RESET(GPIOB, GPIO\_Pin\_12)

#define LED0\_TOGGLE BIT\_REVERSE(GPIOB, GPIO\_Pin\_12)

由于没有使用嵌入式操作系统，所以需要自行实现延时功能。使用for循环能实现不太精确的延时，但由于云台系统的周期更迭会影响到控制精度，故需要使用STM32的滴答定时器Systick，实现较为精确的系统延时功能。

#define LED1\_ON BIT\_SET(GPIOB, GPIO\_Pin\_13)

#define LED1\_OFF BIT\_RESET(GPIOB, GPIO\_Pin\_13)

#define LED1\_TOGGLE BIT\_REVERSE(GPIOB, GPIO\_Pin\_13)

Systick是一个24位的系统节拍定时器，具有自动重载和溢出中断功能，节拍时间可以配置。使用Systick实现延时功能，本质上是制造一个循环，通过传入一个延时变量作为判断条件，在Systick中断中不断改变此变量，当变量值到一定条件时，则退出循环，实现如下。

static \_\_IO uint32\_t SystickDelayedTime;

void Delay\_us(uint32\_t delayedTime){

SystickDelayedTime = delayedTime;

SysTick->CTRL |= SysTick\_CTRL\_ENABLE\_Msk;

while(SystickDelayedTime != 0);

}

void CalledBySystickHandler(void){

if(SystickDelayedTime > 0) SystickDelayedTime -= 1;

}

void SysTick\_Handler(void){

CalledBySystickHandler();

}

在实现云台系统过程中，避免不了调试，故实现了一个串口，用于系统调试信息打印和调试参数的输入。作为调试串口的为USART1，波特率配置为115200bps，数据位8位，停止位1位，无奇偶校验和流控制。输入使用的是USART1的接收中断，在中断函数中将接收到一个无符号字符型变量存储到接收数组中；输出通过重写fputc()函数，将printf()函数打印重定向到串口，通过调用printf()函数便能将调试信息从串口打印出来。输入输出实现如下。

void USART1\_IRQHandler(void){

if(USART\_GetITStatus(USART1, USART\_IT\_RXNE) == SET){

usart1RxBuff[cntUsart1RxBuff] = USART\_ReceiveData(USART1) & 0xff;

cntUsart1RxBuff = (cntUsart1RxBuff+1) % USART1\_RX\_LEN;

}

}

1. 系统电压与手控输入

int fputc(int ch, FILE \*f){

USART\_SendData(USART1, (uint8\_t)ch);

while(USART\_GetFlagStatus(USART1, USART\_FLAG\_TXE) == RESET);

return (int)ch;

}

系统电压和手控输入的摇杆部分均是通过ADC采集数据，不同的是，系统电压使用的是ADC2，并且通过ADC的转换结束中断来更新数据；而摇杆部分则是通过DMA（Direct Memory Access，直接存储器访问）功能将ADC1转换的数据存储到静态数组中，减轻了CPU的负担。ADC2中断更新数据实现如下，ADC1和ADC2共用一个中断向量和中断函数。

void ADC1\_2\_IRQHandler(void){

ADC2\_Value = ADC\_GetConversionValue(ADC2);

}

两种ADC采集方式采集回来的数据，均使用一阶数字低通滤波器处理，系统电压使用的滤波器的实现如下。

Lipo\_Voltage = calWeight \* Lipo\_Voltage +

(1-calWeight) \* (ADC2\_Value\*0.0061767578125);

手控输入的按键输入使用的是扫描方式。由于按键普遍存在抖动问题，故加入了消抖功能。按键输入的整体思想为，当按键按下时间小于N个系统周期时，认为按下无效；若大于N个系统周期时，则判断按键弹起时间是否大于N个周期，若小于则认为弹起无效，重新等待按键弹起，否则则认为完成了一次按键操作。具体实现如下。

if(cntKeyDown < KEY\_DOWN\_PERIOD){

if(GPIO\_ReadInputDataBit(GPIOC, GPIO\_Pin\_2) == 0x00) cntKeyDown += 1;

else cntKeyDown = 0;

}else if(cntKeyDown < KEY\_DOWN\_PERIOD\*2){

if(GPIO\_ReadInputDataBit(GPIOC, GPIO\_Pin\_2) == 0x01) cntKeyDown += 1;

else cntKeyDown = KEY\_DOWN\_PERIOD;

}else{cntKeyDown = 0;

Joystick\_Z ^= 0x01;

}

1. 姿态更新与电机控制

姿态数据由主控器通过I2C方式读取两个MPU6050数据解算获得，两个传感器均使用模拟I2C方式与主控器通信。模拟I2C的关键在于控制信号的高低电平来仿真I2C信号，因此，分别将引脚I/O配置、拉高/低电平、读取电平操作抽象成宏，实现如下。

#define I2C1\_SDAin \

{ \

i2c1GPIOInitStruct.GPIO\_Pin = GPIO\_Pin\_7; \

i2c1GPIOInitStruct.GPIO\_Mode = GPIO\_Mode\_IN\_FLOATING; \

GPIO\_Init(GPIOB, &i2c1GPIOInitStruct); \

}

#define I2C1\_SDAout \

{ \

i2c1GPIOInitStruct.GPIO\_Pin = GPIO\_Pin\_7; \

i2c1GPIOInitStruct.GPIO\_Mode = GPIO\_Mode\_Out\_OD; \

GPIO\_Init(GPIOB, &i2c1GPIOInitStruct); \

}

#define I2C1\_SCL(x) GPIO\_WriteBit(GPIOB, GPIO\_Pin\_6, (BitAction)(x))

#define I2C1\_SDA(x) GPIO\_WriteBit(GPIOB, GPIO\_Pin\_7, (BitAction)(x))

#define I2C1\_SDA\_READ (GPIO\_ReadInputDataBit(GPIOB, GPIO\_Pin\_7) & 0x01)

MPU6050姿态结算中，俯仰角和滚转角使用了一阶互补滤波进行数据融合，俯仰角解算的部分实现如下。

Mpu6050\_Pitch=calWeight\*atan2(Mpu6050\_Accel\_Y, Mpu6050\_Accel\_Z)\*180/MATH\_PI+ (1-calWeight)\*(Mpu6050\_Pitch+Mpu6050\_Gyro\_X\*calPeriod);

姿态更新数据作为PID控制器的输入，在具体实现位置式PID控制器时，加入了变积分的思想，即，设法改变积分项的累加速度，使其与偏差大小相对应，偏差越大，积分越慢，否则，积分越快。变积分思想具体实现如下。

if(fabs(et) >= ERROR\_LIMIT\_UPPER) indexICal = 0.0;

else if(fabs(et) <= ERROR\_LIMIT\_LOWER) indexICal = 1.0;

else indexICal = 1 - ((fabs(et)+ ERROR\_LIMIT\_LOWER)/ ERROR\_LIMIT\_UPPER);

dI = indexICal \* pitchIPara \* et\_sum;

PID控制器计算输出的控制量控制无刷电机转速的原理，是通过控制驱动电机的SPWM波的变化周期实现，正弦函数值静态数组生成和SPWM波生成实现如下。

## ****下载调试****

pwmSin[cntL] = (uint16\_t)( ( sin( (cntL+1)\*1.0/sineArraySize\*2\*MATH\_PI )+1 )

\*TIM\_PERIOD/2 );

mPitch.stepA = (mPitch.stepA + sineArraySize + mdir\_cal\_factor[mdir]) % sineArraySize;

STM32全系列芯片都带有ISP（In-System Programmability，在系统可编程）功能，即可以使用上位机软件通过串口来对STM32芯片内部的存储器进行编程，编程后设备可自动运行程序，进入正常的工作状态。

STM32F103RCT6的ISP功能使用的是USART1，若上位机无串口，则需要使用USB转UART线，连接上位机与STM32。同时，将芯片BOOT0引脚连接到VCC，BOOT1引脚连接到GND，便将芯片启动方式设置为ISP模式，即内部存储器模式。

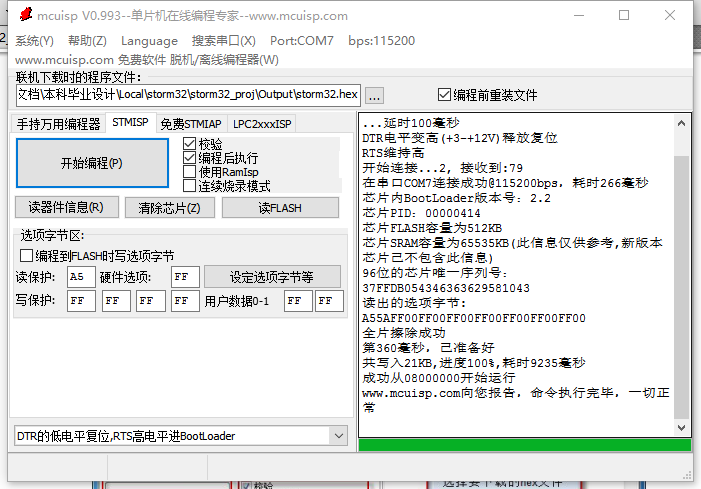
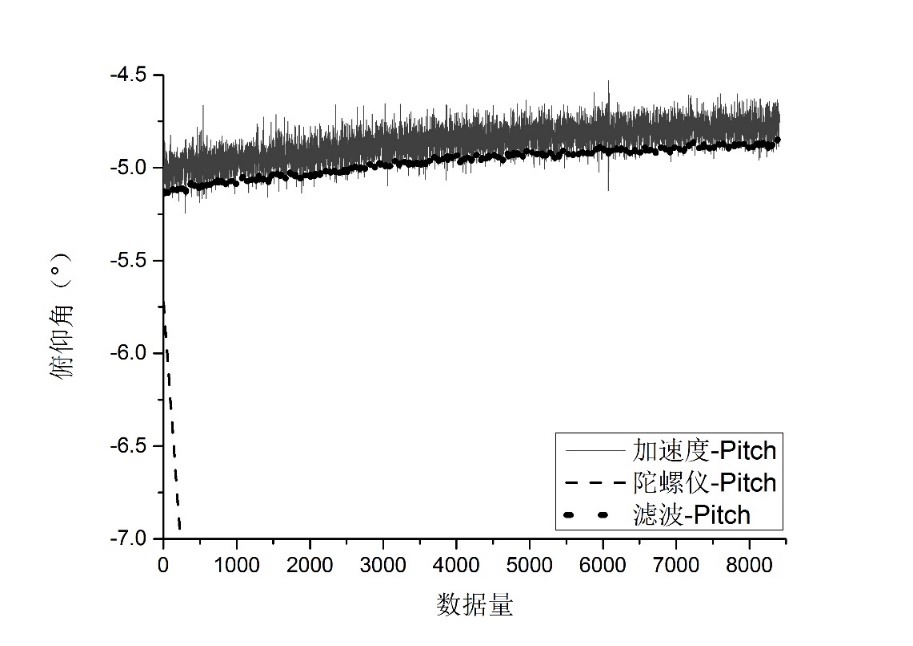
上位机使用的ISP下载工具为MCUISP，需要选择的有串口、波特率和MDK软件编译生成的后缀为\*.hex的十六进制文件。程序下载完成后软件状态如图6-5所示。下载完成后BOOT0和BOOT1引脚都要连接到GND，芯片才能进入Flash模式，从Flash读取程序并执行。

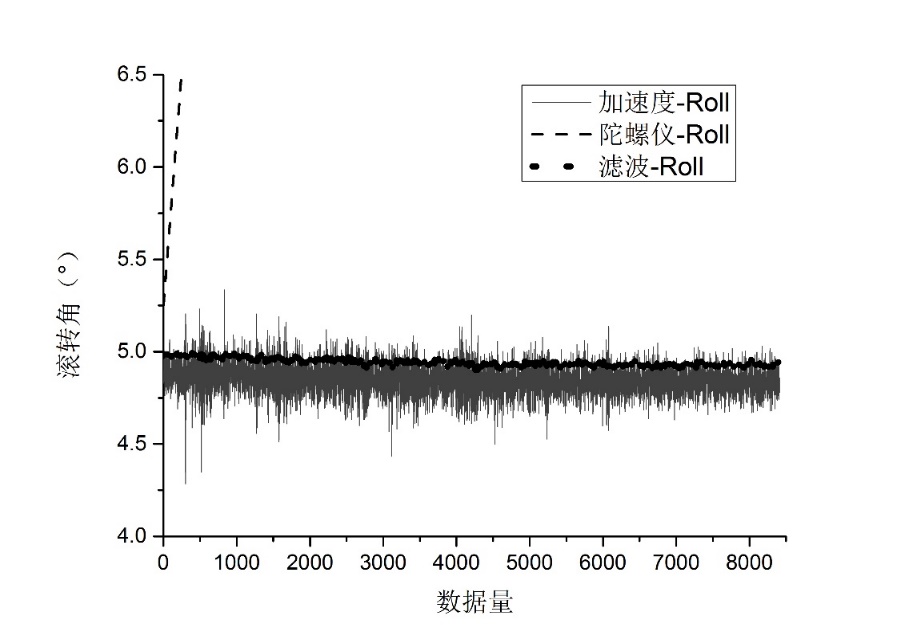
图6-5 一种ISP下载工具

USART1同时也是调试串口，它与上位机的连接方式同使用ISP模式时一致，它的串口功能与ISP功能是分时使用的。上位机通过串口工具（如XCOM）便能接收单片机程序发送的调试信息和发送调试参数给单片机，串口工具的参数配置需要同单片机程序设置的调试串口的参数一致。

# 系统测试

## ****俯仰角和滚转角****

**系统姿态解算使用的是一阶互补滤波，为了测试滤波效果，将 MPU6050静置于台面上，将从调试串口获得的数据绘图处理。图7-1和图7-2分别为加速度计、陀螺仪和一阶互补滤波计算所得的俯仰角和滚转角的比较。**

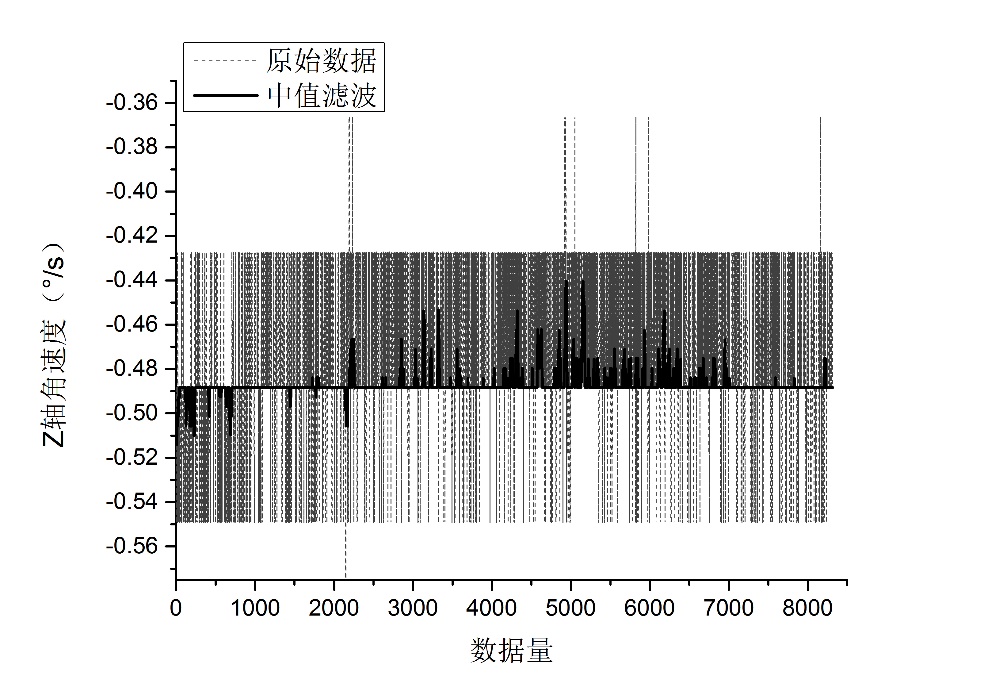
**图7-1 俯仰角对比**

**图7-2 滚转角对比**

**由两对比图可以看出，一阶互补滤波效果显著。没有经过校准的陀螺仪角度积分误差极大，只依靠加速度计算出来的角度则噪声极大。**

## ****积分用Z轴角速度****

姿态传感器Z轴角速度存在的主要问题之一是噪声大，故使用了中值滤波。**将MPU6050静置于台面上，将从调试串口获得的数据绘图处理。**中值滤波前后Z轴角速度对比如图7-3所示。

图7-3 中值滤波前后Z轴角速度对比

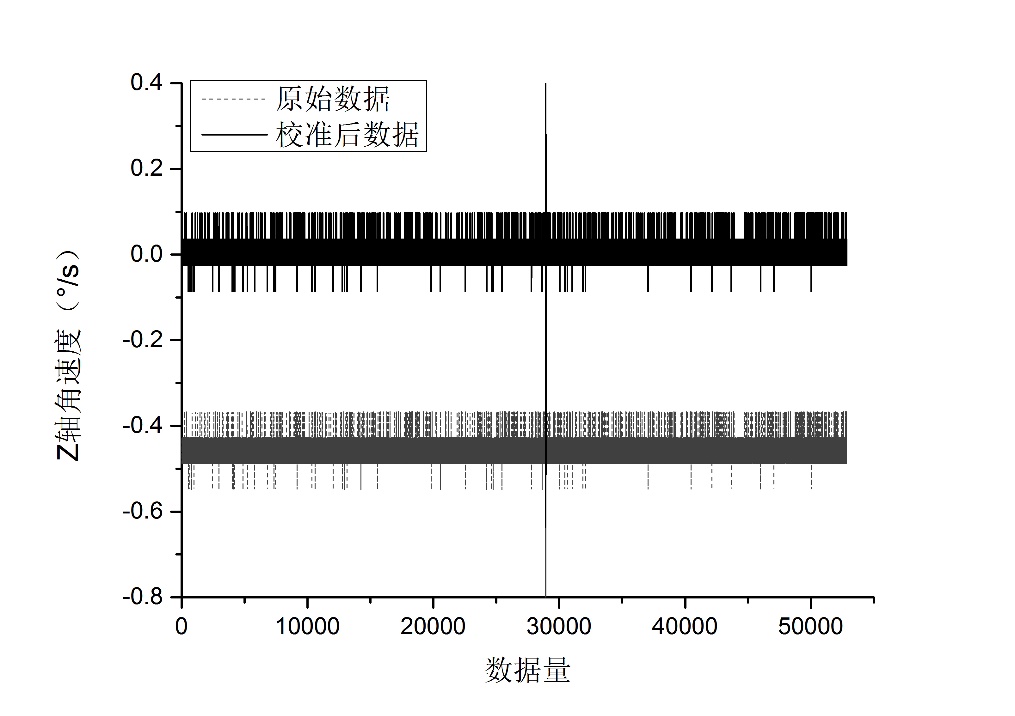
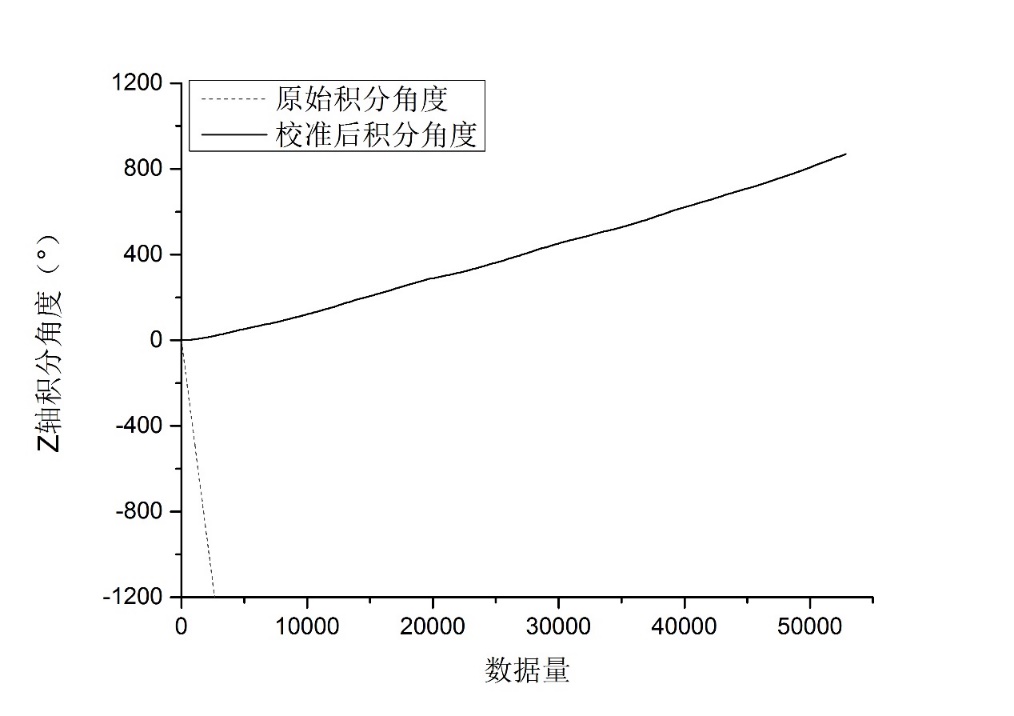
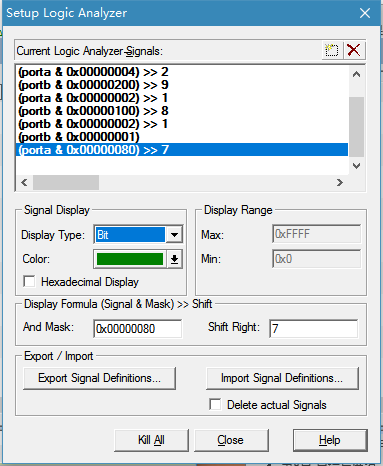
使用中值滤波后，能有效抑制数据波动，消除一定噪声。不难发现，滤波后滤波后的数据与实际状态存在一定偏差，这个偏差是由陀螺仪传感器本身的特性引起的。所以在进行滤波等操作之前，对Z轴角速度做了一个偏差校准，主要是消除零点漂移问题。零点漂移校准前后的Z轴角速度对比如图7-3所示，Z轴积分角度对比如图7-4所示。

图7-3 零点漂移校准前后Z轴角速度对比

图7-4 零点漂移校准前后Z轴积分角度对比

校准后的Z轴角速度回到零点附近，波动范围在-0.1°至0.1°之间，零点漂移基本去除。Z轴积分角度校准后均仍存在其他漂移情况，但去除零点偏移后积分角度的漂移趋势明显变缓。

## ****无刷电机SPWM波****

**控制无刷电机使用的SPWM波由STM32定时器生成，共九个通道。使用MDK软件的仿真调试功能，可以在不需要将程序烧写到芯片上的情况下，实现对程序的调试。在其中的逻辑分析仪工具中，添加定时器九个通道对应的引脚，并将显示类型设置为BIT，如图7-5所示。执行代码，便可以在逻辑分析仪中观察到定时器各通道生成的SPWM波，如图7-6所示，其中的正弦波为后期图片处理以辅助理解。**

**图7-5 MDK逻辑分析仪工具**

**图7-6 STM32定时器生成SPWM波**

## ****锂离子电池电压****

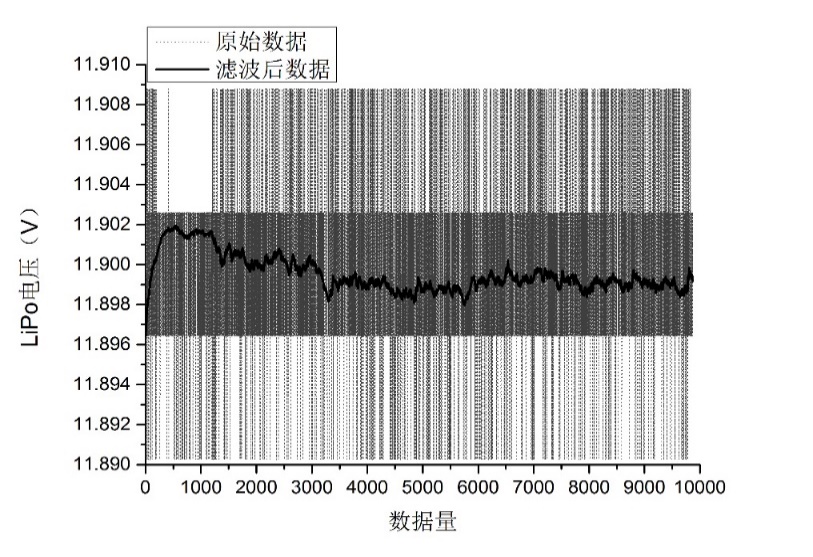
LiPo电压使用ADC采集，然后使用一阶数字低通滤波处理。使用稳压模块作为系统测试电压输入，使用调试串口存储打印数据，将滤波前后的LiPo电压数据进行绘图对比，如图7-6所示。

图7-6 滤波前后电压对比

显然，滤波后的电压数据精度有明显提高，数据波动范围在±0.03V左右，波动极小。同时，滤波器消除了大量的数据噪声，提高了数据的稳定性和可靠性。

# 总结与展望

## ****全文总结****

本课题在研究了国内外云台技术基础之上，设计和搭建了以STM32单片机为核心器件的三轴手持云台。设计过程紧跟微惯性陀螺仪的发展，分析了云台控制算法和电机的运动控制原理，并结合实际安装使用要求进行结构设计和软件编写，经多次调试，云台基本达到设计要求。本课题在设计实现过程中主要成果和收获如下：

1. 使用了低成本的单片机、IMU、电机等， 结合3D打印结构，成功搭建了性能稳定的云台系统。
2. 研究了自动控制中的常用的经典PID控制算法，结合姿态角的一阶互补滤波，实现了云台自稳控制量实时准确地计算。
3. 使用SPWM波控制无刷直流电机转动，可根据要求改变SIN函数采样点数量调整电机控制精度。

## ****课题展望****

本课题研究的自稳云台系统，基本达到了预期的效果。但出于时间和研究水平约束，云台设计中还有一些存在的问题和改进的地方，主要包括：

1. 主控制器部分，在适当时候关闭部分功能时钟可降低能耗；采用嵌入式实时操作系统uC/OS，能改进系统周期的准确性，提高数据滤波等需要系统周期作为系数的操作的精度，同时，多任务也能方便云台系统的模块的增删和管理。
2. 姿态结算部分，可以采用中间计算量较少的四元数算法，可以避免欧拉角的奇异性和万向节死锁问题，但在理解上需要多花点时间。
3. 姿态控制部分，可以对各轴建立精确的仿真数学模型进行分析，提高对云台的控制精度。
4. 云台结构部分，可以使用耐高温的打印耗材打印电机座，或者使用激光烧结技术打印金属电机座，自行设计的电机座能大幅度减小云台的体积。同时，可以在横滚轴上增加配重块，用来平衡不同相机的重量，使云台未上电初始化时已接近水平状态，能有效减轻电机负载和减少发热量，延长电机寿命。

参考文献

[1] 郭阳志. 基于ARM的增稳云台姿态控制系统研究[D]. 北京: 湖北工业大学, 2015.

[2] 聂水茹. 高动态环境下捷联惯导的优化算法研究[D]. 西安: 西北工业大学, 2003.

[3] 郭炳坤. 无人机三轴稳定云台智能控制算法的研究[D]. 厦门: 集美大学, 2013.

[4] 王成波,戈振扬,贺勇,等. 基于STM32F4的机载三轴云台控制器设计[J]. 中国农机化学报, 2016, 37(8): 177-181.

[5] 邱云平,伍宝玉. MPU-6050模块角度算法处理及在嵌入式中的应用[J]. 江西科技学院学报, 2014, 9(2): 32-35.

[6] 陈孟臻. 基于MPU-6050陀螺——加速度仪三轴云台的研究[J]. 百色学院学报, 2015, 28(3): 132-137.

[7] 龙达峰. 基于椭球拟合的三轴陀螺仪快速标定方法[J]. 仪器仪表学报, 2013, 34(6): 101-107.

[8] 陈孟臻. 三轴WiFi摄像头舵机云台的设计与实现[D]. 北京: 广西大学, 2015.

[9] 李青,王青. 3D打印：一种新兴的学习技术[J]. 工业设计, 2013, 29(4): 29-35.

[10] 吴芬芬. 3D打印物体的稳定平衡优化[D]. 北京: 中国科学技术大学, 2016.

致谢

时光荏苒，犹如白驹过隙，四年大学学习生活已接近尾声。在这四年时间里，有三年时间，是在黄培灿老师的指导下学习、成长。本课题也是在黄老师的亲自指导下完成作品的设计和论文的撰写。黄老师是一位学术作风严谨、项目经验丰富、理论功底深厚、非常具有亲和力的老师。在与黄老师相处的这三年里，我不仅收获了理论知识及动手实践能力，也更让我懂得为人处事的道理，这将是我一生的财富。在此，特向尊敬的黄老师表达最崇高的敬意和最衷心的感谢！

其次，要感谢创客空间的胡天明师弟，在作品的结构的打印部分，为我提供了3D打印机，并帮助我学习打印机的配置和使用。

最后，感谢我的朋友和家人，总是默默地给予我生活上的关怀和学业上的支持，是我前进的动力和坚实的后盾。