

文章编号: 2095-2163(2020)03-0279-06

中图分类号: TP368.1

文献标志码: A

基于树莓派的四轴飞行器设计

田东波, 徐鹏程, 张德俊, 周琦, 杨云露

(上海工程技术大学 机械与汽车工程学院, 上海 201620)

摘要: 传统四轴飞行器只能通过远程遥控的方式工作, 在实际运用中存在很大的局限性。针对上述问题本文提出并设计了一款基于树莓派的四轴飞行器, 其与传统的飞行器相比多了一个智能控制模块, 该模块可以代替人工控制。飞行器采用 STM32F103 作为飞控模块的核心芯片, 采用树莓派 3B 模块作为智能控制模块, 通过 MPU9250 模块获取飞行器的姿态信息, 通过 GPS 模块获取飞行器的位置信息, 通过 WiFi 或 4G 实现对飞行器的远程遥控。手机或 PC 端通过基于 TCP/IP 的协议向飞行器的智能控制模块发送控制指令从而实现对飞行器的控制, 而智能控制模块通过 SPI 总线以及相关协议对飞控模块进行控制。实验测试表明, 本文所提出的四轴飞行器设计方案具有可行性, 值得进一步研究。

关键词: 四轴飞行器; 树莓派; STM32; TCP/IP; SPI

Design of four-axis aircraft based on Raspberry Pi

TIAN Dongbo, XU Pengcheng, ZHANG Dejun, ZHOU Qi, YANG Yunlu

(School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

[Abstract] The traditional four-axis aircraft can only work by remote control, which has great limitations in practical use. In response to the above problems, this paper proposes and designs a four-axis aircraft based on the Raspberry Pi. Compared with the traditional aircraft, it has an intelligent control module, which can replace the manual control. The aircraft adopts STM32F103 as the core chip of the flight control module, adopts the Raspberry Pi 3B module as the intelligent control module, acquires the attitude information of the aircraft through the MPU9250 module, acquires the position information of the aircraft through the GPS module, and realizes the remote control of the aircraft through WiFi or 4G. The mobile phone or PC transmits control commands to the intelligent control module of the aircraft through a TCP/IP-based protocol to control the aircraft, and the intelligent control module controls the flight control module through the SPI bus and related protocols. The experimental test shows that the design scheme of the four-axis aircraft proposed in this paper is feasible and worthy of further study.

[Key words] quadcopter; Raspberry Pi; STM32; TCP/IP; SPI

0 引言

四轴飞行器最初被发明是用于军事领域, 随着近些年半导体以及嵌入式行业的发展, 四轴飞行器的制作成本以及开发难度大大降低, 其应用逐渐走向民用领域。在日常生活中, 四轴飞行器最常见的用途是航拍, 而此类用途对飞行器的功能要求并不是特别高, 仅需要支持无线遥控。随着近些年网购的兴起, 四轴飞行器甚至被用来运送快递, 在此类应用中四轴飞行器仅仅支持无线遥控远远是不够的, 还需要其具有智能控制功能。传统的四轴飞行器仅仅采用一块 MCU 芯片作为控制单元的核心, 无法满足相关智能控制算法运行的硬件要求, 因此需要进一步研究和开发。

1 相关理论

1.1 四轴飞行器工作原理

四轴飞行的机身布局通常有 2 种模式, 分别是十字模式和 X 模式。在实际应用中, 四轴飞行器通

常会携带摄像头等视频图像采集设备, 为保证视线的良好, 四轴飞行器大多采用 X 模式的机身布局。X 模式的机身布局如图 1 所示。

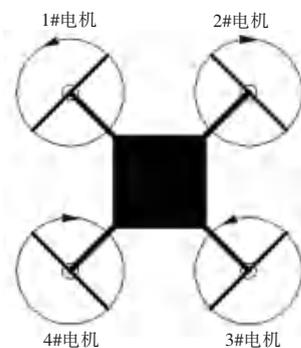


图 1 X 模式机身布局图

Fig. 1 X mode body layout

四轴飞行器通过 4 个旋翼产生的升力来飞行, 而为了保证受力的平衡, 通常采用一对正桨和一对反桨, 图 1 中 1#和 3#采用一对正桨, 2#和 4#采用一

基金项目: 上海工程技术大学研究生科研创新项目 (E3-0903-19-01157)。

作者简介: 田东波 (1993-), 男, 硕士研究生, 主要研究方向: 电池热管理; 徐鹏程 (1994-), 男, 硕士研究生, 主要研究方向: 汽车电子控制技术、嵌入式系统开发技术。

收稿日期: 2019-11-22

对反桨。接着根据图 1 进一步阐述四轴飞行器的 8 种运动状态的原理^[1]：

(1) 垂直向上或向下运动：在四轴飞行器稳定后,1#、2#、3#、4#电机以相同的速率同时加速,飞行器就会垂直向上运动,反之则会垂直向下运动。

(2) 向左或向右运动：在四轴飞行器稳定后,1#、4#电机转速下降并且 2#、3#电机转速提高,飞行器向左运动,反之飞行器向右运动。

(3) 向前和向后运动：在四轴飞行器稳定后,1#、2#电机转速下降并且 3#、4#电机转速提高,飞行器向前运动,反之飞行器向后运动。

(4) 顺时针改变航向和逆时针改变航向：在四轴飞行器稳定后,1#、3#电机转速下降并且 2#、4#电机转速提高,飞行器顺时针改变航向,反之飞行器逆时针改变航向。

1.2 飞行姿态表示

飞行器的姿态表示有多种方法,最常用的为欧拉角表示方法,即为翻滚角、倾仰角和偏航角。假设全局坐标系 F_w 固定与地面保持静止,而随飞行器转变的坐标系 F_b 初始姿态与 F_w 是重合的为 F_{b0} ,然后 F_{b0} 按如下顺序旋转至最终的姿态 F_b ^[1-6]：

(1) F_{b0} 绕着 F_w 的 Z 轴按右手坐标系规则转动 ψ 角,即为偏航角,并假设新姿态为 F_{b1} 。

(2) F_{b1} 绕着 F_{b1} 的 Y 轴按照右手坐标系规则转动 θ 角,即为倾仰角,并假新姿态为 F_{b2} 。

(3) F_{b2} 绕着 F_{b2} 的 X 轴按照右手坐标系规则旋转 ϕ 角,即为翻滚角,此时转动到最终坐标系 F_b 。

根据上述过程可得最终的旋转矩阵为：

$$R = \begin{pmatrix} \hat{c}_\theta \hat{c}_\psi & -C_\phi S_\psi + S_\phi S_\theta C_\psi & S_\phi S_\psi + C_\phi S_\theta C_\psi & \ddot{y} \\ \hat{c}_\theta S_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & -S_\phi C_\psi + C_\phi S_\theta S_\psi & \dot{y} \\ \hat{s}_\theta & S_\phi C_\theta & C_\phi C_\theta & \ddot{z} \\ \hat{c}_\theta - S_\theta & & & \ddot{z} \end{pmatrix} \quad (1)$$

其中, C_x 为 $\cos x$, S_x 为 $\sin x$ 。

1.3 闭环控制及 PID 控制算法

四轴飞行器是通过控制 4 个旋翼的转速来实现四轴飞行器的不同运动状态,其对系统稳定性的要求非常高,一般采用闭环控制系统。而闭环控制系统中最常用的控制器是 PID 控制器, PID 控制器可以根据比例环节 P、积分环节 I 以及微分环节 D 来调整输入值,从而使得控制更加精确以及系统更加稳定。PID 控制器的比例环节 P 对应于当前误差,积分环节 I 对应于累计误差,微分环节 D 对应于预测的未来的误差,实际应用中通过调节这三个参数

来设计满足需求的控制系统。在计算机应用领域通常使用数字式 PID, 数字式 PID 算法可表示为^[1-3,7]：

$$u_k = k_p \left(e_k + \frac{T}{T_i} \sum_{j=0}^k e_j + T_d \frac{e_k - e_{k-1}}{T} \right). \quad (2)$$

其中, $e(k)$ 为第 k 个采样时刻的误差角; T 为采样周期; k_p 为比例环节系数; T_i 为积分环节系数; T_d 为微分环节系数。

2 总体设计方案

2.1 系统设计方案

四轴飞行器的系统设计方案如图 2 所示。本设计采用树莓派 3B 模块作为智能控制模块,并且通过 UART 接口在树莓派上扩展 GPS 模块以及通过 USB 接口扩展摄像头模块;采用 STM32F103 作为飞控模块的控制芯片,该芯片分别对 4 个电调模块提供 PWM 输出,并且通过 I2C 接口扩展 MPU9250 模块。手机或 PC 端通过 TCP/IP 来向四轴飞行器的智能控制模块发送控制指令,智能控制模块通过 SPI 向飞控模块下发飞行状态指令。

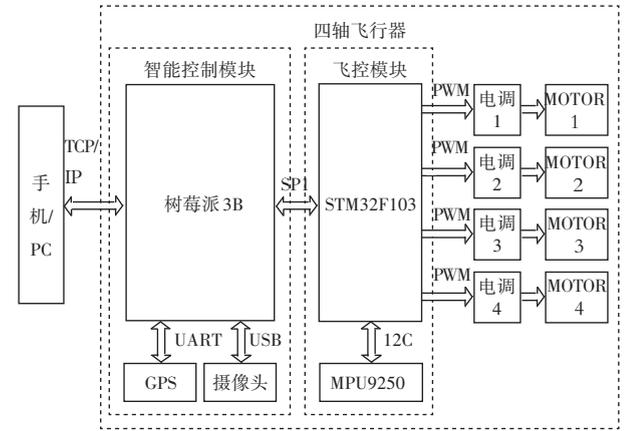


图 2 系统设计方案框图

Fig. 2 System design block diagram

由图 2 可知,对于系统中各主要模块的设计功能可做概述如下:树莓派 3B 模块是英国树莓派基金会开发的一款卡片式电脑,该模块内部集成有高性能的 SOC 芯片以及 WiFi、蓝牙等通信模块,除此之外该模块还具有丰富的外设接口以及供二次开发的 GPIO 接口^[8-9];GPS 模块采用的是移远科技的 L80R 模块;STM32F103 芯片是意法半导体的一款 32 位 ARM 架构的 MCU;MPU9250 是一款高性能的位姿传感器模块,内部集成有 3 轴陀螺仪、3 轴加速度计以及 3 轴磁力计。

2.2 电源设计方案

本设计采用 16.8 V 的锂电池作为输入电源,同

时为满足不同模块的供电要求,采用 DCDC 先将 16 V 降压到 5 V,然后再通过 DCDC 将 5 V 降压到 3.3 V。16 V 降 5 V 采用的是 MP2225 芯片,该芯片可以最大提供 5 A 的电流输出;5 V 降 3.3 V 采用的是 AMS1117-3.3,该芯片最大提供 1 A 的电流输出。系统的供电方案如图 3 所示。

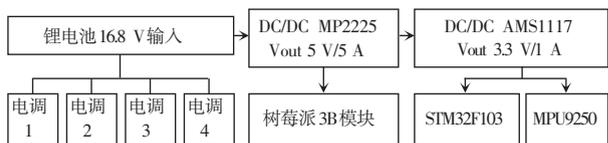


图3 系统供电方案树状图

Fig. 3 System power supply scheme tree

3 硬件设计

四轴飞行器的硬件设计部分包括:电源电路设计、STM32F103 最小系统设计、电机驱动电路设计、MPU9250 电路设计等。这里,对其中主要电路的设计拟展开研究论述如下。

3.1 电源电路设计

16.8 V 降压到 5 V 采用的是 MP2225 电源芯片,5 V 降压到 3.3 V 采用的是 AMS1117-3.3 电源芯片,电源电路设计如图 4 所示,图 4 中 PWR_5V_EN 与树莓派相连用于控制系统电源的通断。

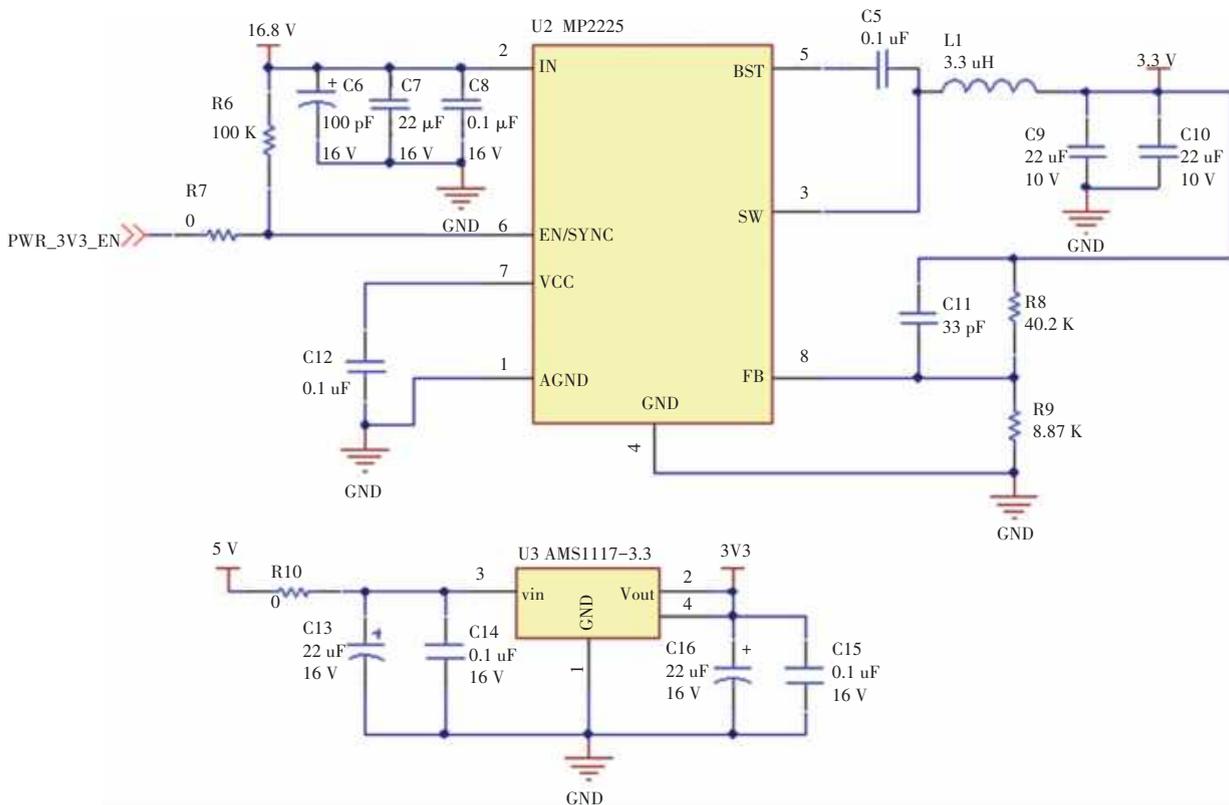


图4 电源电路设计图

Fig. 4 Power circuit design

3.2 STM32F103 最小系统设计

STM32F103 的最小系统设计如图 5 所示。该芯片共有 48 个引脚,采用 3.3 V 直流供电,由一个 8 MHz 的无源晶振提供主时钟以及一个 32 768 Hz 的无源晶振提供实时时钟,14、15 是 SPI 接口引脚,21、22 是 I2C 接口引脚,31、32 是 UART 接口引脚,42、43、45、46 引脚是 PWM 输出引脚。

3.3 MPU9250 电路设计

MPU9250 的电路设计如图 6 所示,该模块采用 3.3 V 直流供电,MPU9250_SDA 是 I2C 接口的数据

线,MPU9250_SDL 是 I2C 接口的时钟线,与 STM32F103 芯片的 I2C 接口相连。

4 软件设计及测试

本文设计的飞行器与传统的四轴飞行器相比,增加了智能控制模块。智能控制模块的主要功能是接收和解析用户通过手机或 PC 端发送的指令并执行相应的控制程序,通过 SPI 向飞控模块发送控制指令实现对四轴飞行器的运动控制。为实现上述功能,需指定相关的协议,本节主要对相关协议的制定、软件的设计以及测试做详细探讨与描述。

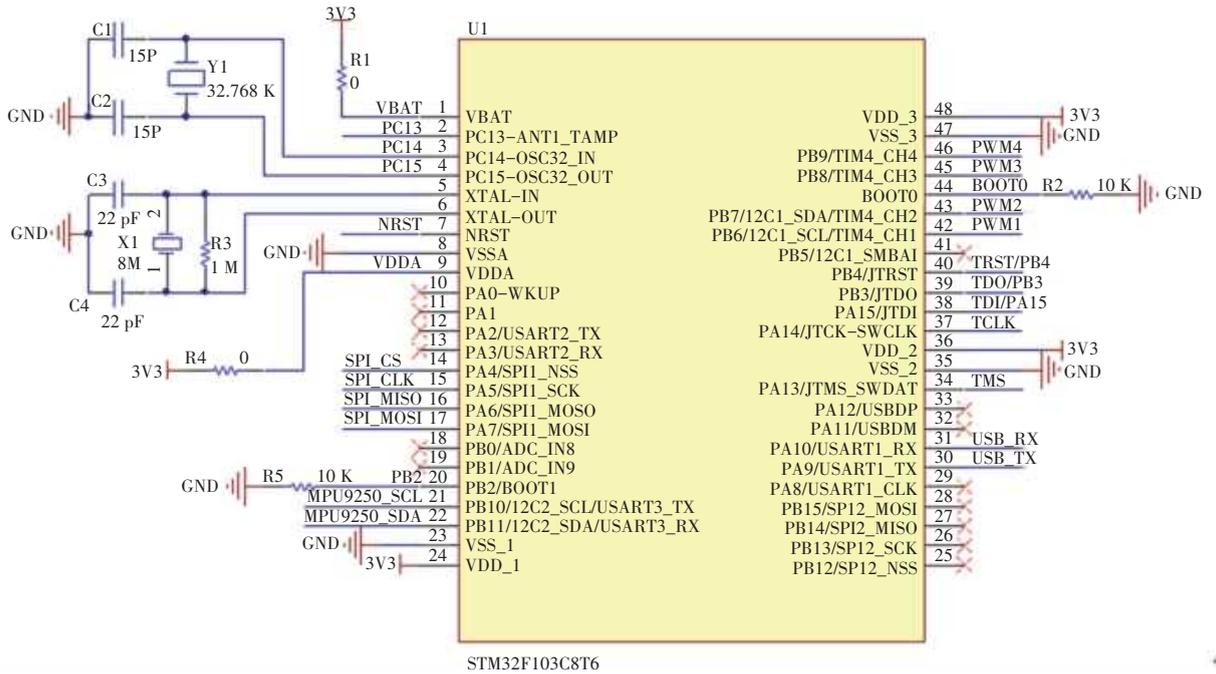


图5 MCU最小系统电路设计图

Fig. 5 MCU minimum system circuit design

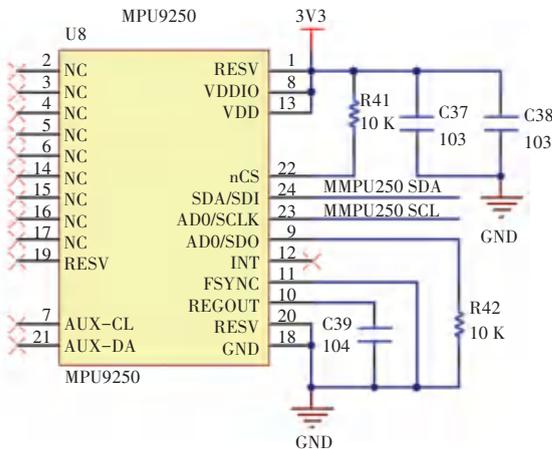


图6 MPU9250电路设计图

Fig. 6 MPU9250 circuit design

4.1 相关通信协议的制定

4.1.1 基于TCP/IP的控制协议

手机或PC端对四轴飞行器的控制协议是基于TCP/IP传输层协议的透明传输协议,一条报文的定义见表1。由表1分析可知,一条报文中各组成部分的含义可做阐释分述如下。

表1 控制指令报文定义

Tab. 1 Control instruction message definition

报文头	指令	设备号	指令参数	CRC 校验	报文尾
2	1	2	8	1	2

(1) 报文头: 一条报文的起始标志, 约定为“0xbee”。

(2) 指令: 四轴飞行器的控制指令, 取值在0x00~0xff之间, 最多可表示255条控制指令, 具体见表2。

(3) 设备号: 用于区别不同的设备。

(4) 命令参数: 控制指令相关的参数, 具体定义见表2。

(5) 报文尾: 一条报文的终止标志, 约定为“0xfeed”。

表2 指令及参数定义

Tab. 2 Instruction and parameter definition

指令	指令定义	参数定义
0x00	基本控制指令	字节0: bit0~4依次表示飞行器的4类运动状态(上下、前后、左右、旋转), bit5~7未定义值为0; 字节2~7未定义值均为0x00;
0x01	定向飞行指令	字节0~1: 终点经度; 字节2~3: 终点纬度; 字节4~5: 起点经度(可不设置默认值为0x0000); 字节6~7: 起点纬度(可不设置默认值为0x0000)
0x02	指定区域盘旋	字节0~1: 中心点经度; 字节2~3: 中心点纬度; 字节4~7: 飞行半径(单位m)
0x03~0xff	暂未定义	暂未定义

四轴飞行器的回复报文定义见表3, 由于篇幅有限, 回复报文的详细定义不再赘述。

表 3 回复报文定义

Tab. 3 Reply message definition					字节
报文头	报文类别	设备号	数据	CRC 校验	报文尾
2	1	2	8	1	2

4.1.2 基于 SPI 的控制协议

智能控制模块通过 SPI 向飞控模块发送指令, 控制飞行器的飞行状态(控制指令)、获取飞行器的状态参数(读指令)、设置相关参数(写指令)。控制指令和读指令的长度都是 2 个字节、共 16 位, 指令

表 4 基于 SPI 的指令定义

Tab. 4 SPI-based instruction definition

指令	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
控制指令	0	0	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0	0
读指令	0	1	A4	A3	A2	A1	A0	0	0	0	0	0	0	0	0	0
写指令	1	0	A4	A3	A2	A1	A0	D8	D7	D6	D5	D4	D3	D2	D1	D0

4.2 通信软件的设计

根据所制定的协议对相关通信软件进行设计, 软件设计主要包括树莓派中的命令解析执行软件以及 STM32F103 中指令解析执行程序。由于篇幅有限, 本小节只对 MCU 中相关程序设计做详细介绍。研究中采用 SPI 接收中断来处理相关指令, 中断的处理程序如下:

```

u16RecData = SPI_I2S_ReceiveData ( SPI1 ); // 读取数据帧
cmdFrame.cmd = ( u16RecData & 0xC000 ) >> 14; // 解析命令
switch( u8Cmd )
{
case CMD_CONTROL: // 控制指令
cmdFrame.cmd_C_data = ( u16RecData & 0x3E00 ) >> 9; // 解析控制参数
vControl( cmdFrame.cmd_C_data ); // 执行控制程序
break;
case CMD_READ: // 读指令
cmdFrame.cmd_RW_address = ( u16RecData & 0x3E00 ) >> 9; // 解析读地址
SPI_I2S_SendData ( SPI1, u16GetData ( cmdFrame.cmd_RW_address ) ); // 返回读的结果
break;
case CMD_WRITE: // 写指令
cmdFrame.cmd_RW_address = ( u16RecData & 0x3E00 ) >> 9; // 解析写地址
cmdFrame.cmd_W_data = u16RecData &

```

的定义见表 4。bit15-14 是指令位, 用来区分不同的指令。控制指令中 bit13 以及读写指令的 bit8 都是奇偶校验位, 用来进行指令帧数据的校验。控制指令中 bit12-8 是指令参数位, 其中 bit12 代表“启/停”, bit11 代表“顺/逆”, bit10 代表“左/右”, bit9 代表“前/后”, bit8 代表“上/下”。读写指令中 bit13-9 是读写地址位, 用来区分不同的参数。写指令 bit7-0 为写数据位, 用来存放写入的数据。

```

0x01FF; // 解析写数据
vWriteData ( cmdFrame.cmd_RW_address, cmdFrame.cmd_W_data ); // 执行写操作
break;
default:
break;
}

```

4.3 软件测试

为进一步验证本设计方案的可行性以及通信协议的可靠性, 本文进行了软件测试, 软件测试的过程如图 7 所示。

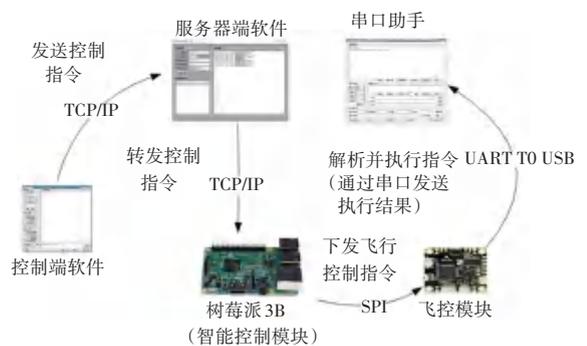


图 7 软件测试过程示意图

Fig. 7 Schematic diagram of the software test process

控制端软件通过 TCP/IP 协议将控制指令发送给设备管理服务器端, 服务器端再将命令转发给设备, 设备在接收到控制指令后解析并执行相关操作, 本测试中为更好地体现命令解析和执行的结果, 将设备端的执行结果通过串口发送到 PC 端。测试结果表明, 飞行器设备能够很好地执行控制端软件发送过来的指令, 达到了预期的效果。