

2022 年山东省大学生电子设计竞赛

送货无人机（B 题）



摘 要

本次比赛设计了一个基于 STM32H743VIT6 作为主控芯片的四旋翼飞行器的送货无人机，通过与协处理器 STM32F103C8T6 的交互完成吊舱、语音、键盘等模块的实现。其中包括了摄像头模块 openmv，姿态解算模块 ICM20602，定点模块采用光流定点，测距模块则采用超声波，气压计 SPL06 以及激光测距。送货无人机能够根据要求，在识别给定的特定区域特征模板后，能对图中区域进行搜寻并且悬停，在指定位置放下吊舱，同时播放指定语音，而后返回起点平稳降落。同时，也能使用摄像头模块识别功能完成钻呼啦圈，主要涉及路径规划算法，四元数姿态解算算法，卡尔曼滤波算法。

关键词：无人机；送货；识别；路径规划

送货无人机（B 题）

【本科组】

一、系统方案

本系统主要由单片机控制模块、摄像头模块、定点模块、姿态解算模块、定高模块、电源模块组成，下面分别论证这几个模块的选择。

1、主控制器件的论证与选择

1.1 单片机模块

方案一：选择 STM32F103RBT6 作为主控芯片，这款芯片属于意法半导体常见的芯片之一，资料众多，操作简单，常用于嵌入式开发设计之中。但是对于无人机这类需要高性能数据处理与控制能力的系统而言并非最优选择。

方案二：选择 STM32H743VIT6 作为主控芯片，这款芯片主频明显高于方案一所选芯片，且 PWM, UART 等资源充足，性能极强，完全可以满足对飞行器的控制需求。

综合以上两种方案，我们选择了方案二。

1.2 姿态解算模块

方案一：采用 MPU6050 进行姿态的解算，通过 I2C 进行通信，最快的通信速度为 400k/s，可以得到当前无人机的四轴数据。

方案二：采用 ICM20602 进行姿态的解算，通过 SPI 的方式进行通信，最快通信速率为 10M/s，传感器噪声也大幅降低，可以更加高效地得到无人机当前的运动状态。

综合以上两种方案，为了得到更好地控制效果，我们选择了方案二。

1.3 定高模块

方案一：采用超声波测距的方式定高，超声波测距模块的测距范围为 0~150cm，精度为 3mm，精度达到 0.3%，测距符合要求，但受外界环境影响变动大，在飞行过程中不是十分稳定。

方案二：采用气压计 SPL06 定高，气压计数据波动较大，精度低，室内定高效果可能略差。

方案三：采用激光测距的方式定高，测距范围 0.1m~12m，准确度 1%，抵抗环境光性能较强，更安全稳定，但激光易受高度突变的影响，可与加速度计数据项融合得到实际高度。

综上，我们将三种方案进行了结合，即激光定高，加速度计与气压计数据修正，超声波测距同时发挥作用，对高度进行更精准的把控。

1.4 定点模块

方案一：二维激光雷达定点。使用二维激光雷达获得无人机相对周围环境的距离。将激光雷达数据与 IMU 数据融合用于无人机定点。经测试在室内定点效果较好。但是由于场地周围有网且比较稀疏，雷达扫描效果不好，难以用于定点。

方案二：光流定点。由于室内比赛没有 GPS 信号，光流定点成为首选方案，并与 IMU 数据融合，可以达到较好的定点效果。光流所具备的最大优点是无须了解场景的信息，就可以准确地检测识别运动目标位置，且在摄像机处于运动的情况下仍然适用。

综合以上两种方案，我们选择了方案二。

1.5 视觉模块

方案一：采用树莓派和 opencv 库实现视觉识别

树莓派（Raspberry Pi）是一个小型电脑，在树莓派上部署基于 python 的 opencv 程序能够对于传输的图片进行比较全面的图像操作，但是调试操作难。

方案二：采用 k210 实现视觉识别

K210 专为机器视觉与机器听觉多模态识别而设计，降低了学习 AI 视觉的门槛。但是对于此题来说，使用训练 ai 模型的方法无法获得准确坐标，对训练集的要求比较高。

方案三：采用 openmv 实现视觉识别：

openmv 的机器视觉模块，搭载了 MicroPython 解释器，使其可以在嵌入式端进行 python 开发。openmv 的集成度高，函数调用方便，逻辑清晰，而且实际运用上效果好。

综合以上三种方案，选择方案三。

二、系统理论分析与计算

1、前往目标地点

通过计算实际场地尺寸的精确数据以及距各目标点的期望距离，我们进行了飞行函数的标定。因现场因素的不确定性，我们进行了多次调试，可以保证到达坐标基本准确。此外，为了保证目标的准确性，我们做了双重保证。先通过光流以及传感器的精确数据测量，飞到目标点周围的位置，在通过视觉引导返回坐标的偏移值，对飞行器位置进行了进一步标定，从而确保了位置不会出现较大偏差。

2、通过串口通信完成多设备交互执行任务

当 openmv 识别到与目标位置的偏移值小于一定阈值时，我们通过串口的联通，利用自定义的协议发送数据包，使语音模块播放对应语音，且吊舱缓慢降落，实现送货的模拟。

2、摄像头识别图形的计算

(1) 识别三角形

在识别过程中，采用三角形色块大小与颜色框大小的比值作为判断依据，正方形圆形的比值要相对大于三角形的比值，可以判断出形状。其中识别色块的过程实现颜色识别。

(2) 识别正方形和圆形

采用 openmv 自带的正方和圆形的检测函数识别颜色，再采用色块识别判断坐标

3、openmv 像素坐标系与厘米坐标系的转换的计算

openmv 中，使用了 QQVGA 的大小，识别的像素点为 180×160 ，其中，openmv 识别到的是在摄像头中显示出来的像素坐标，而控制飞机时，需要传入厘米坐标的参数，为此，我们经过了一系列调试，先做了消除畸变处理，之后，我们根据 openmv 镜头的范围对实际区域进行了量取，并确定了转换系数。进行了调试、转换之后，发现识别效果较好。

三、电路与程序设计

1、电路的设计

(1) 系统总体框图

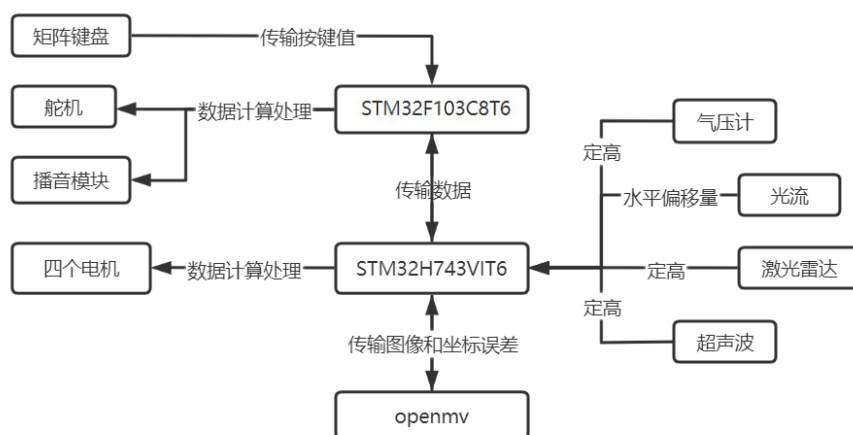


图 1 系统总体框图

(2) 硬件电路整体框图

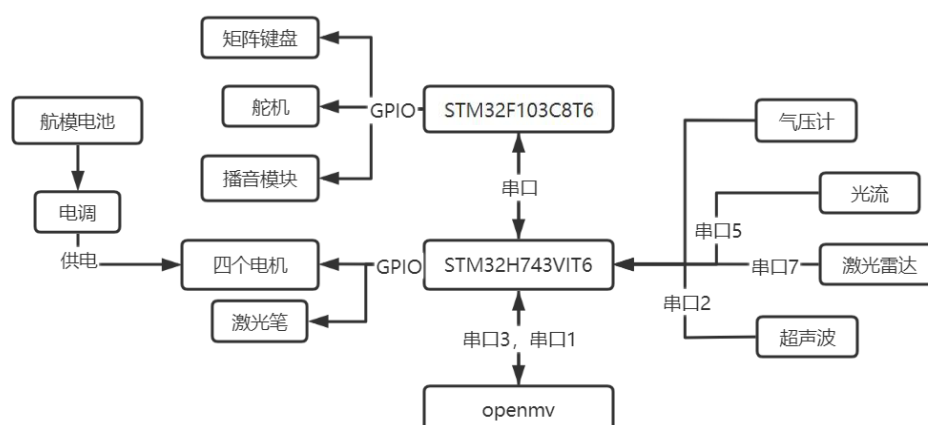


图2 硬件电路整体框图

2、程序的设计

(1) 程序功能描述与设计思路

(1) **openmv** 识别数字和形状，能够返回坐标值。三角形主要通过识别色块的框与色块比例完成识别，圆形是利用霍夫圆变换检测，方形是利用四元检测算法来识别矩形。颜色主要通过色块检测中的阈值来实现。

(2)先通过交互传给飞控目标位置,前往大致坐标后利用 openmv 传回的纠偏信息,对具体位置进行微调,来保证悬停位置的精确性。悬停后通过串口通信,执行播放语音以及释放吊舱功能。

(2) 程序流程图

1、主程序流程图

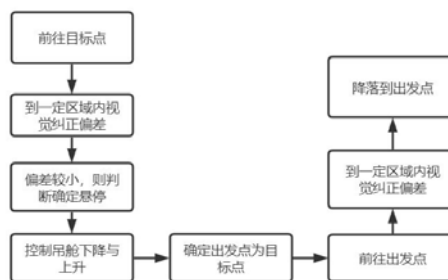


图 3 主程序流程图

2、摄像头程序流程图:

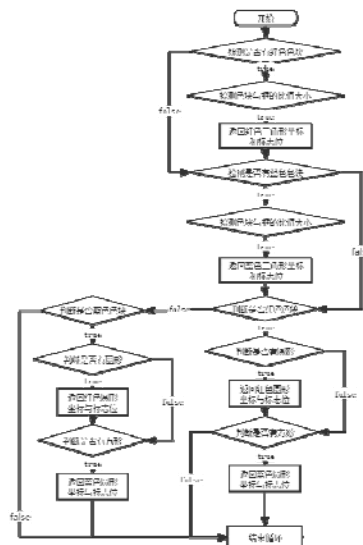


图 4 摄像头程序流程图

四、设计算法

4.1 四元数姿态解算

欧拉角转化成四元数： Φ, θ, ψ 分别对应 roll, yaw, pitch 的旋转角度

$$q_0 = \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \quad (1.1)$$

$$q_1 = \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \quad (1.2)$$

$$q_2 = \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \quad (1.3)$$

$$q_3 = \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} - \sin \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \quad (1.4)$$

四元数便于计算，欧拉角便于观察和处理数据。

四元数与过渡矩阵的变换

$$C_b^a = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_3q_2 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_3q_2 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (1.5)$$

一阶龙格库塔法求四元数：

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_{t+\Delta t} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_t + \begin{bmatrix} -\omega_x \cdot q_1 - \omega_y \cdot q_2 - \omega_z \cdot q_3 \\ \omega_x \cdot q_0 - \omega_y \cdot q_3 + \omega_z \cdot q_2 \\ \omega_x \cdot q_3 + \omega_y \cdot q_0 - \omega_z \cdot q_1 \\ -\omega_x \cdot q_2 + \omega_y \cdot q_1 + \omega_z \cdot q_0 \end{bmatrix} \quad (1.6)$$

根据四元数进行姿态解算。

4.2 卡尔曼滤波

卡尔曼滤波一般分为两个步骤，状态预测和更新。利用状态外推方程、协方差外推公式、状态更新方程、协方差更新方程、卡尔曼增益五个公式，可以有效进行滤波，使传入的实时数据更精确。

基本公式如下：

1. 预测阶段

$$\mathbf{x} = (\mathbf{F} * \mathbf{x}) + (\mathbf{B} * \mathbf{u})$$

$$\mathbf{P} = (\mathbf{F} * \mathbf{P} * \mathbf{F}^T) + \mathbf{Q}$$

2. 修正阶段

$$\mathbf{y} = \mathbf{z} - (\mathbf{H} * \mathbf{x})$$

$$\mathbf{S} = (\mathbf{H} * \mathbf{P} * \mathbf{H}^T) + \mathbf{R}$$

$$\mathbf{K} = \mathbf{P} * \mathbf{H}^T * \mathbf{S}^{-1}$$

$$\mathbf{x} = \mathbf{x} + (\mathbf{K} * \mathbf{y})$$

$$\mathbf{P} = (\mathbf{I} - (\mathbf{K} * \mathbf{H})) * \mathbf{P}$$

3.模拟数据效果

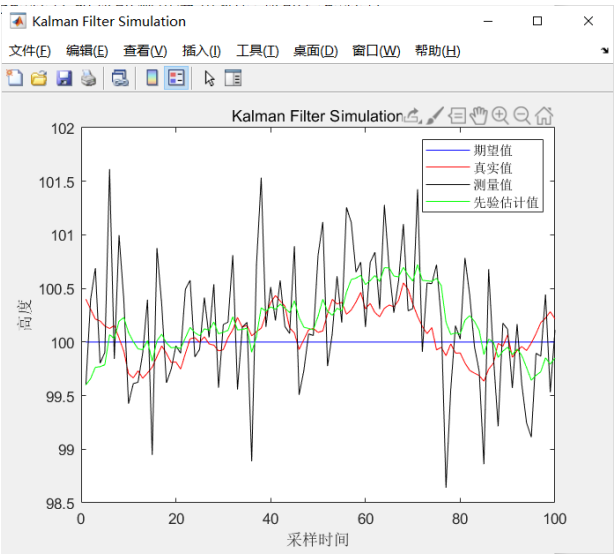


图 5 采样数据与 KLM 滤波数据对比图

五、路径规划

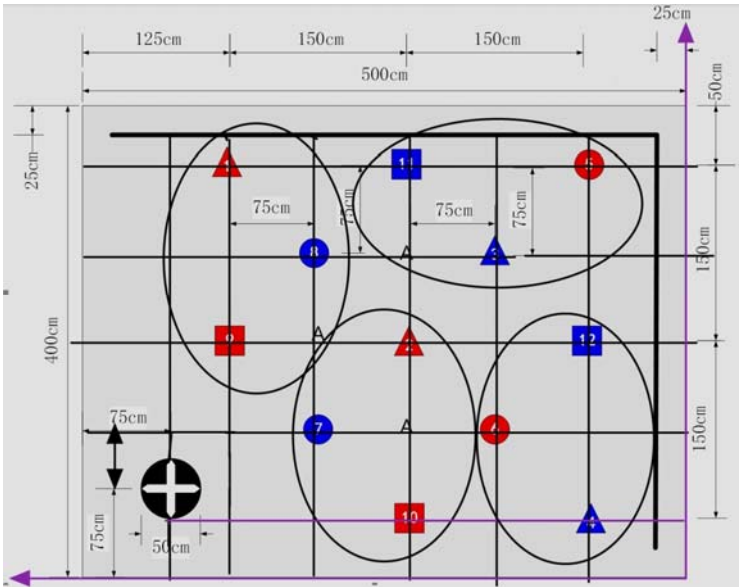


图 6 地图分区图

我们对区域进行了统一划分，此种划分方式的优点为：查找方便，利用分块的方式，不用采用遍历图的方法进行路径挑选。经测试，此分区方案效果较好。

六、测试

1、测试方案

- (1) 手持 openmv 模块，通过 USB 串口与 PC 连接，调用其官方 IDE 对实际

拍摄到并对处理后的画面进行观察和数据分析，能在电脑端观察到清晰的图像，在 PC 端观察目标区域颜色判决结果，openmv 模块工作正常。

(2) 将无人机目标点设为各不同点，进行依次试飞，到达效果较为完好。

2、测试条件

我们严格按照比赛条件制作场地，在测试阶段可通过遥控器观察高度，空间位置，各姿态数据，遥控器数据等，当出现问题时有万用表，示波器等检测，从而方便进行测试。

3、测试结果及分析

(1) 测试结果(数据)

路径	到达目标点时间 (s)	悬停时间 (s)	返回时间 (s)	到达目标点的偏移程度 (cm)	返回起始点的偏移程度 (cm)
8 -> 9	7	7	10	5	10
9 -> 8	6	6	8	8	6
8 -> 7	8	6	11	4	7
7 -> 2	7	6	7	8	5
2 -> 9	9	7	11	9	7
1-> 9	10	6	10	6	8
9 ->1	8	6	16	9	6

表一 任务一数据

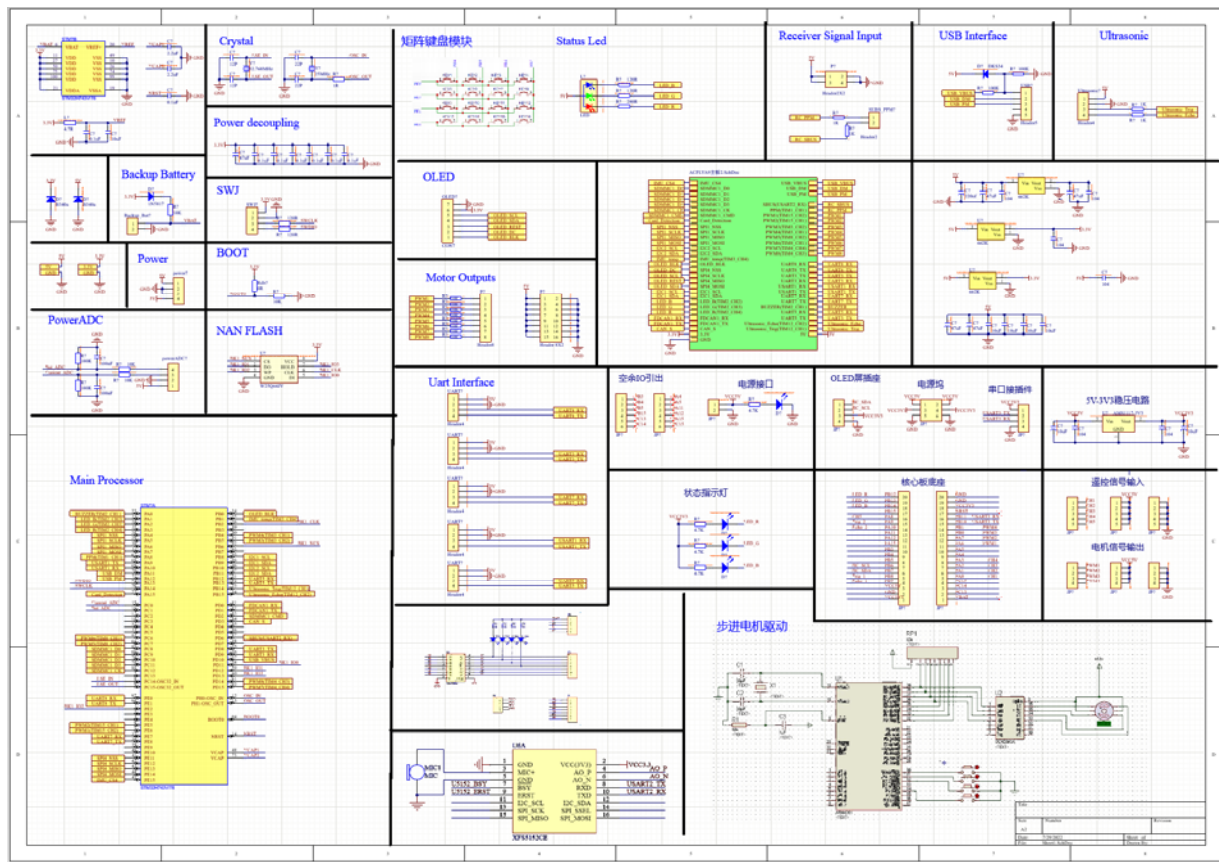
识别模版的类别	找到相应区域与否	寻找时间 (s)	到达目标点的偏移程度 (cm)	返回起始点的偏移程度 (cm)
1	是	15	7	7
2	是	13	5	14
3	是	12	7	9
4	是	10	4	2
5	是	13	6	10
6	是	19	6	12

表二 任务二数据

(2) 测试分析与结论

分析：经过多次测试，飞机能够实现识别指定标识并且准确飞行至指定区域完成送货要求，飞行高度稳定，悬停点与指定区域的误差比较小，并且能够返回起点，完成平稳降落。

附录 1：电路原理图



附录 2：源程序