



Using a 33-bit motor driver cable and Field-Oriented Control (FOC), the RoboMaster C800 Brushless DC Motor Speed Controller enables precise control over motor torque.

Exclusively designed for the RoboMaster M8000 P18 Brushless DC Motor and C800 Brushless DC Motor Speed Controller, this 32-bit Accuspark-18C includes several niches and a thermal sensor.

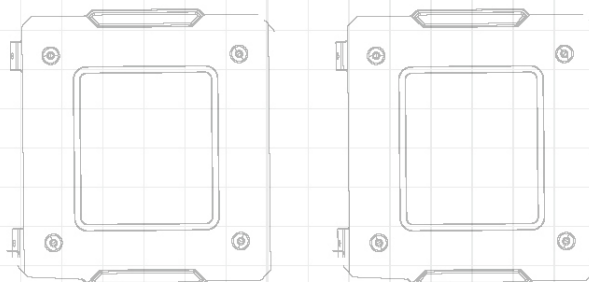
RoboMaster System Breakdown Manual, RoboMaster User Manual, Introduction of RoboMaster System Module

The M8000 Accuspark-18C includes several niches and a thermal sensor, ensuring a complete accurate system when for the RoboMaster robot.

# ROBOMASTER 2021

## GKD战队 高校单项赛 技术报告

RoboMaster2021



# GKD战队飞镖系统技术报告

## 1. 系统概况及说明

在本赛季的备赛过程中，我们对上赛季的设计方案进行了大幅度的改进，其中发射架有原来的多导轨单次同时装填改为单导轨重复装填，参考了许多西安交通大学速加网笃行战队的飞镖发射架开源设计，飞镖也改为了无动力设计。但是由于机器人制作经验不足等问题，飞镖发射架的稳定性一直存在欠缺。但是飞镖在经过多次迭代后已经达到了我们认为的较高水平，不管是集成度，稳定性还是准确性都获得了长足的进步。

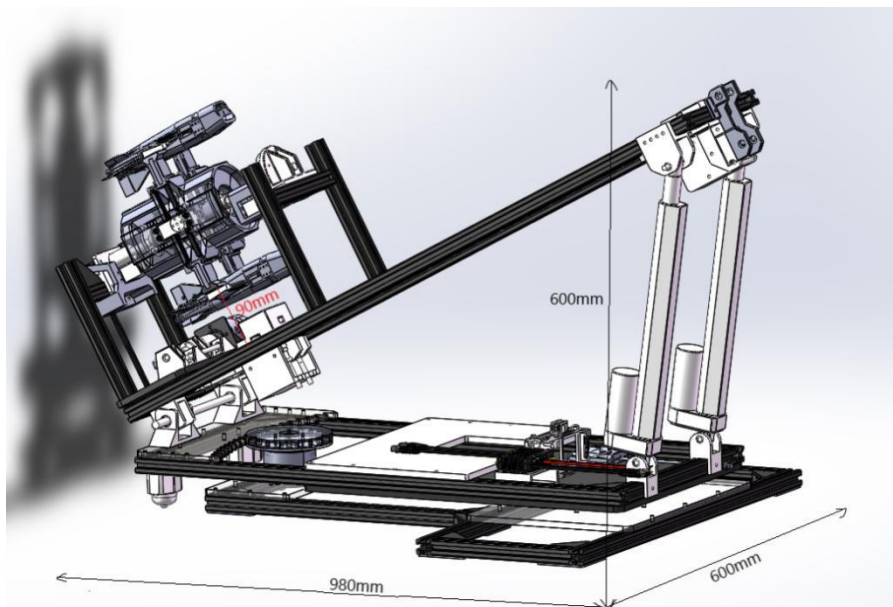
## 2. 发射架部分机械设计方案

### 2.1 设计思路

由于飞镖需要连续打击不同方位、不同距离的目标，所以需要有一个能够绕水平面与垂直面旋转的机构，以及能够自动装填飞镖的结构。即发射架被分为三个主要部分，即可转动底盘，可调角导轨和自动装弹弹鼓。由于飞镖自身并不带动力，所以飞镖的所有动能都需要发射架提供。这就要求发射架需要有至少能够将飞镖投射至前哨站附近的能力。由于飞镖由发射裁判系统指令控制，所以我们使用 A 型开发板获取裁判系统学生串口数据以控制飞镖发射。

### 2.2 参数说明

长：980mm 宽：600mm 高：600mm 重量：17.5Kg 夹爪：90mm（相对于滑轨）



## 2.3 传感器种类

微动开关（2 个）：欧姆龙 V-15-1C25，结构简单，具有微小触点间隙，灵敏度高，寿命长。



陀螺仪（1 个）：JY901B，高精度角度精度，Z 轴航向角加入磁场传感器滤波，解决漂移引起的累计误差，可以长期输出稳定航向角，体积小，支持水平和垂直安装。



## 2.4 执行机构

M3508 电机（1 个）：驱动滑块结构使得橡皮筋完成蓄力，给飞镖提供足够的初速度。



GM6020 电机（2 个）：一个用来驱动齿轮转动结构调整 Yaw 轴角度，另一个驱动弹鼓旋转结构完成飞镖装填。



DS3119 舵机 (7 个)：一个用于控制飞镖释放，两个用于自动装填时飞镖位置调整，四个分别用于夹持单个飞镖进行自动装填。



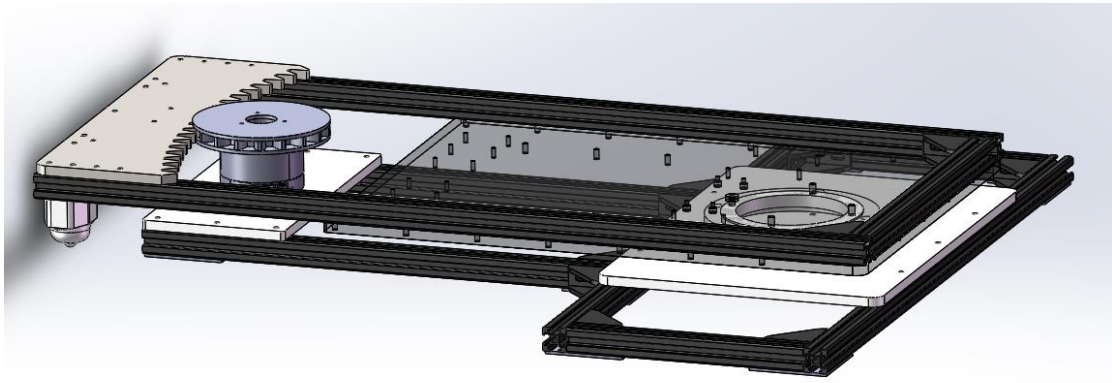
直流推杆电机 (两个)：用于控制发射架导轨角度



## 2.5 安装说明

机械部分安装：所有固定板都是 8mm 的亚克力板，下面每个部分可以分开安装，最后总体安装成为完整的发射架，不存在运输问题。

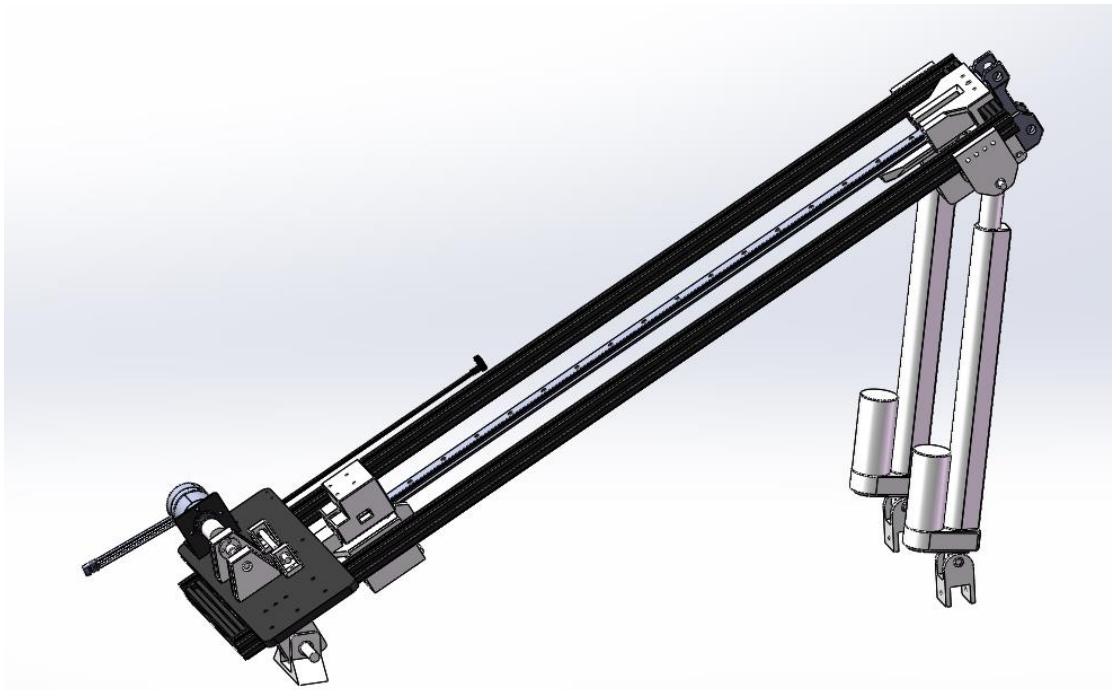
横摆角调整：使用齿轮耦合的思想完成发射架转向的操作，GM6020 电机旋转上面连接的齿轮盘在齿形滑轨上移动，发射架前部使用转子轴承，位于齿形滑轨的圆心。



底盘总成

俯仰角调整：使用两个直线电机连接在发射架底座和滑轨所在的铝方管，连接处使用不锈钢轴和轴承连接，可以随动改变俯仰角，完成设计目标

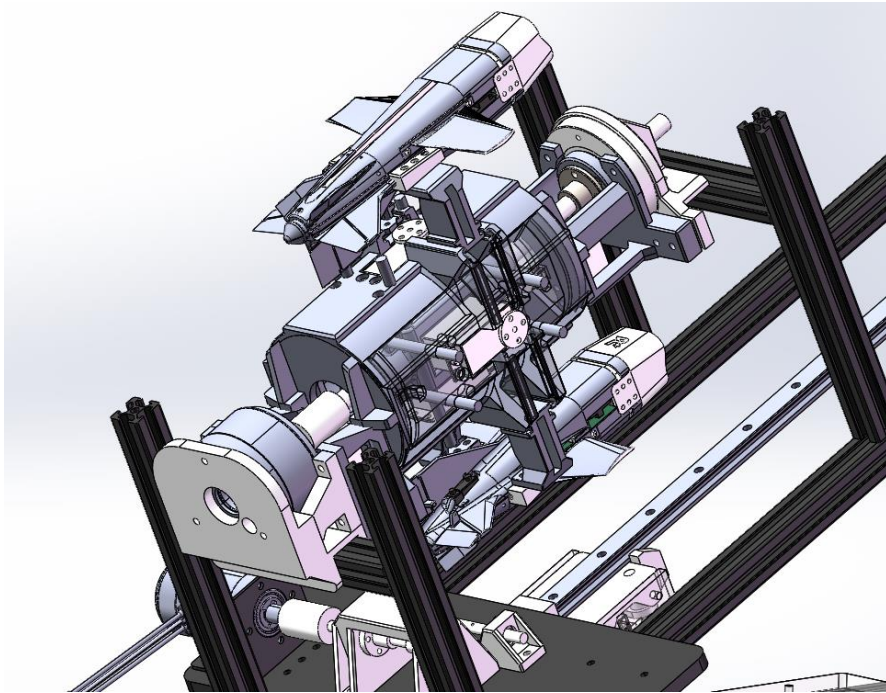
滑块滑行：将不锈钢直线滑轨固定在铝方管支架上，中间最长的铝方管依次安装拉线盒、缓冲 3D 打印件、释放装置和 M3508 电机装置，尾部使用 3D 转接件连接底座，注意安装前后打印件时确保滑块已经安装在滑轨上。



导轨总成

自动装填弹鼓：先安装四根铝方管形成龙门结构，再安装 GM6020 电机和中间旋转结构 3D 打印件，注意所有舵机都要提前装在打印件上。

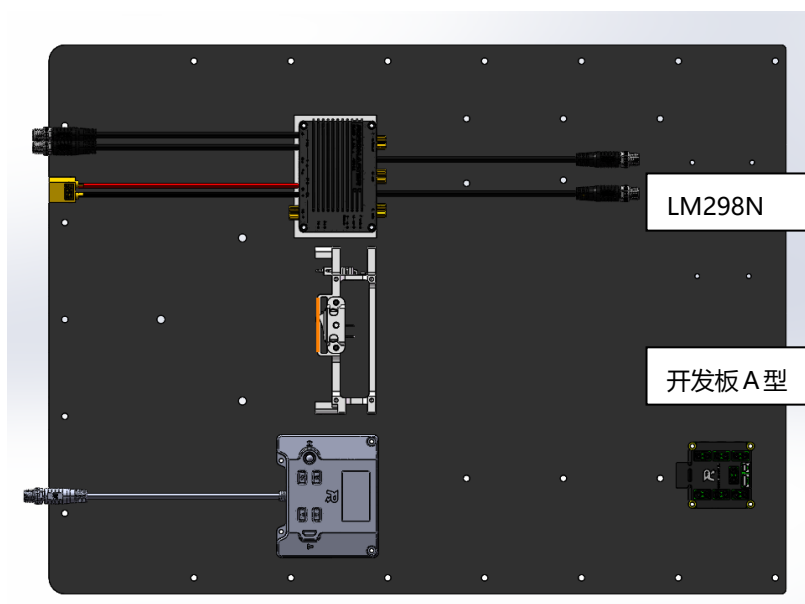




自动装填弹鼓（有两架飞镖及其夹持机构已隐藏）

传感器安装：一个微动开关安装在滑轨前方的缓冲装置上，另一个安装在直线滑轨尾部的 3D 打印转接件上，确定 M3508 电机正转和反转的周期。陀螺仪安装在 M3508 所在的固定板上。

电路板固定安装：在连接底座 Yaw 轴轴承的亚克力板上为裁判系统、电源模块、电调中心板、A 板和电池座固定位置，在所有导线连接完毕后会使用包线管包裹美化，利用扎线带固定，最后在上面覆盖一块黑色亚克力板防止异物进入造成短路。



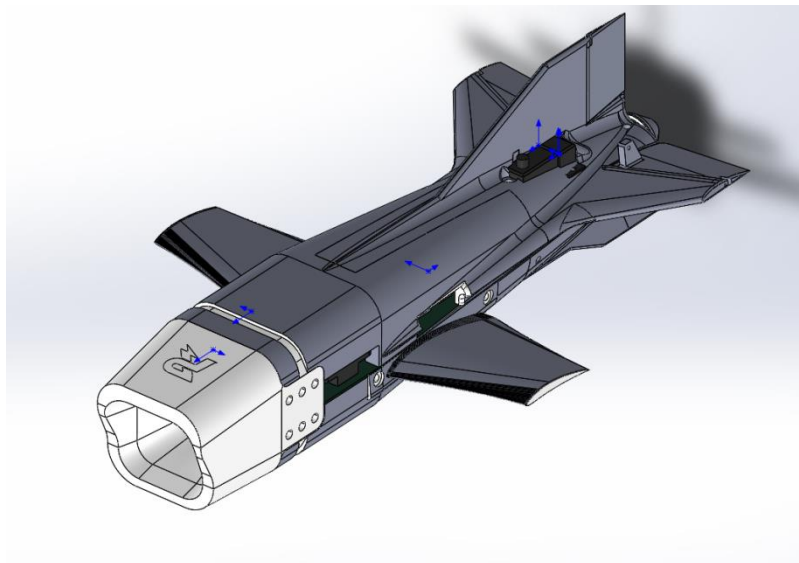
底板布局

## 3. 飞镖部分机械设计方案

### 3.1 设计思路

本赛季中飞镖的外形尺寸限制为了  $200 \times 120 \times 80$ ，不带飞镖头的总重量不超过 150 克，这个尺寸严重限制了飞镖的机翼面积和展弦比，让飞镖无法使用传统低速航模的高展弦比设计。在不考虑使用多层机翼的情况下，如此小面积的机翼几乎不可能产生足够的升力让飞镖进行真正的自主飞行。而重达 20 克的官方飞镖头的加入也说明了这一点，即：飞镖并不是一种“巡航导弹”而更倾向于一种“制导炮弹”。

在这种情况下，飞镖能够使用的气动外形不外乎几种：后掠无尾三角翼，传统布局，鸭式布局，或者纯粹的炮弹（指没有机翼，只有稳定尾翼）。在本赛季的备赛过程中，我们设计并测试了几种不同布局的多架飞镖，最后选择了传统气动布局，具体设计方案及测试视频将于下文说明。



飞镖机器人

### 3.2 主要设计参数

飞镖机器人长 196.5mm，宽 118mm，高 54mm，带飞镖头不带电池重量约为 115 克，带飞镖头带电池重量约为 125 克，尾部有三个舵面以控制飞镖在三个轴向上的运动。

飞镖在待机状态时功率约 0.7W，飞行状态时功率在 5W 左右（包括飞镖头发光功率），一块 7.4V，150mah 电池能够支撑飞镖发射约十次。飞镖 PCB 只使用两个电源滤波电容，总容值约 110uF。支持 5-18V 电池电压输入，电源管

理模块会将电池输入转化为 5V 或 5V 和 3.3V 输出（据 PCB 版本不同输出电压有所差距，具体 PCB 设计将于后文说明）。

飞镖使用了一个 Mini OpenMV 视觉传感器，一个贴片九轴陀螺仪传感器 JY901B 和主控 STM32F103C8T6 自带的 ADC 作为电池电压传感器。

飞镖有三个 1.7 克低压舵机作为执行器。三个舵机分别对应三个舵面，其中俯仰角和滚转角由两个舵面混控，航向角舵面由一个舵机单独控制。

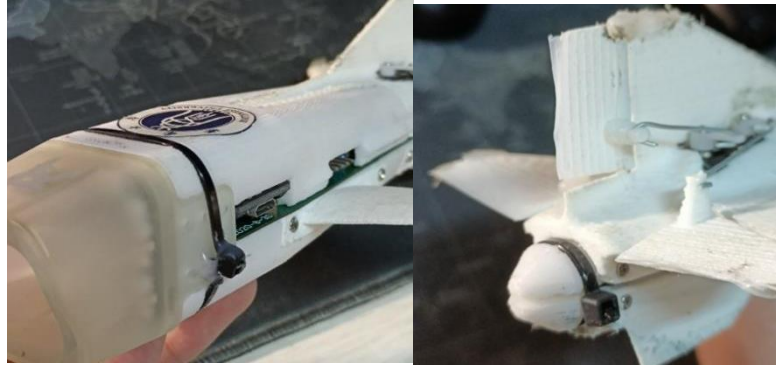
### 3.3 机械设计说明

飞镖的气动外形结合了“炮弹”与“飞机”的特点，既有良好的飞行控制性能，又对气流干扰不敏感。传统布局和静稳定设计让飞镖不易滚转和失速，在单片机不介入的情况下都有着非常好的飞行轨迹。主翼为梯形，采用 NACA6409 翼型。在安装电池后重心处在主翼后缘约 1/3 处，主翼及平尾均有 9° 的上反角，帮助维持飞镖滚转角稳定。平尾后掠且攻角为 -2°。垂尾面积较大，并设计有腹鳍以提升横滚稳定性，并可以帮助抵消在航向角舵面调整时产生的横滚力矩。在视觉不工作时散布区域在半径约 0.8 米的圆内。

飞镖使用 3D 打印的 PLA 和 TPU 材料制作，其中与飞镖头相连接的机身部分使用 PLA，而容易受到碰撞的主翼，水平尾翼和垂直尾翼使用柔性的 TPU 材料制作，在受到强烈撞击的时候提供缓冲，保护使用脆弱 PLA 材料的机身。和 TPU 尾翼一体的柔性的电池舱盖，可以直接翻起更换电池，并通过一个卡扣结构以保证安装电池后的机身一体性和流线外形。

此飞镖具有较好的抗撞击能力，平均在 10 到 15 次发射后才需要进行简单维修（常见故障：舵面脱落，PCB 受撞击位移，舵角损坏）。目前尚未发现能够让飞镖机械结严重损坏的情况。飞镖使用了通用的集成 PCB, JY901B 陀螺仪，nRF24L01 无线电，STM32 主控，蜂鸣器，LM7805 降稳压模块等都直接焊接在 PCB 上。飞镖电路与机械部分相分离，可以通过快速更换集成 PCB 主板进行维修。STLink 通过 PCB 引出到机身侧面，方便程序烧录。机身上下壳之间由 M2 金属螺丝与扎带共同固定（三维图纸中可以看到明显的扎带槽）。





机头和机尾的扎带固定设计

负责视觉的 Mini OpenMV 模块由螺丝直接固定在机身上壳，避免碰撞时损坏摄像头排线。OpenMV 的 USB 和串口都可以直接从飞镖外部连接，避免了程序烧录时拆卸飞镖的麻烦。三个舵机密集安装在机身后部，节省内部空间。舵角通过 0.8mm 细铁丝与舵面相连。舵面则由缝衣针作为转轴安装在尾翼上。部分易损的零件（水平尾翼舵面转轴，飞镖头与机身连接处）在实际测试过程中使用了热熔胶进行补强。



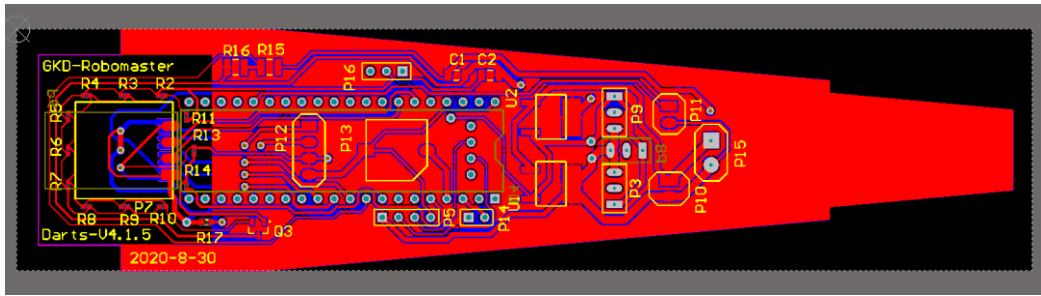
水平尾翼舵面和舵机连杆

### 3.4 电路设计说明（PCB设计说明）

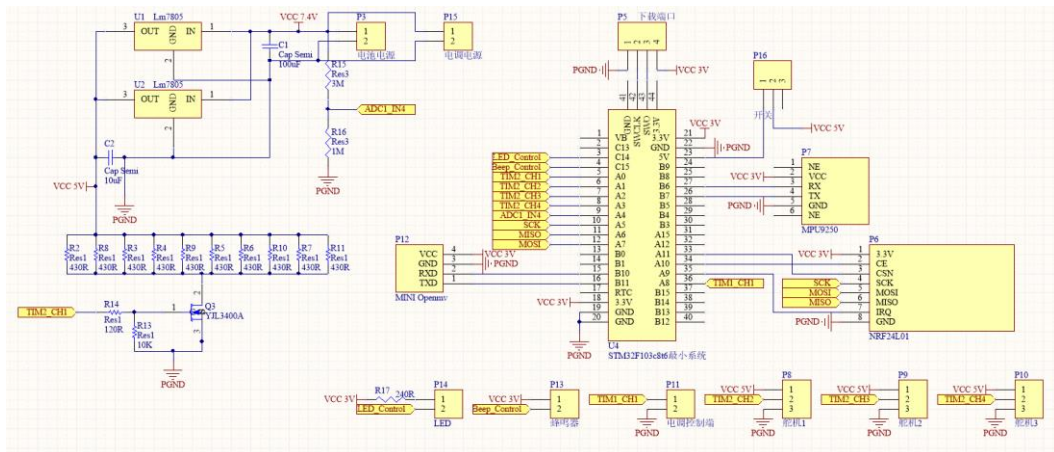
飞镖 PCB 是集成有多种功能的主板，板厚 0.6mm，与飞镖外壳形状一致，直接嵌入在飞镖的上下层机身之间。以 STM32F103C8T6 作为主控，搭载陀螺仪，无线电，蜂鸣器等多种外设，自带稳压及分电功能。陀螺仪周围设计有自加热电路，减少因温漂造成的陀螺仪数据误差。将 STLink 引出，烧录程序时不需要拆解飞镖。并且预留了视觉串口，可以供 OpenMV 与单片机传输数据使用。自带电池电压监测电路，能够实现低电压报警功能。板载的 nRF24L01 2.4GHZ 无线电芯片能够实现无线调参以及飞行状态实时回传。3.3V 有源蜂鸣器和 LED 指示灯可以通过不同的鸣响和闪烁方式提示飞镖的当前状态。

PCB 分为两个版本，分别为 5V 标准舵机和 3.3V 低压舵机设计，其中 5V 版本使用两个 LM7805 降稳压芯片并联作为舵机和单片机部分的供电。而 3.3V 版

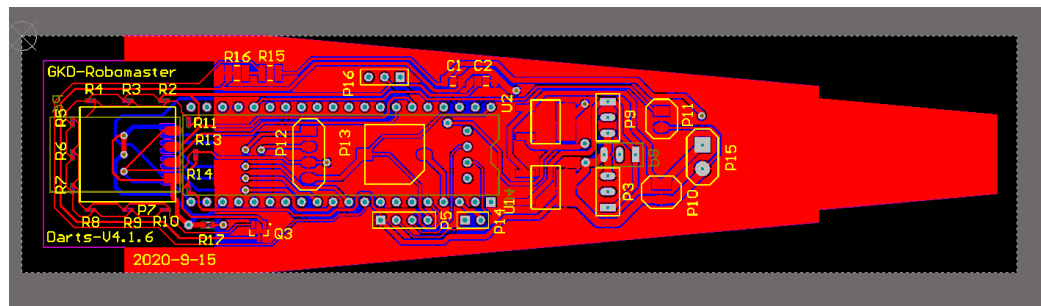
本使用一个 LM7805 和一个 AMS1117 3.3V 分别为单片机和舵机供电。PCB 也预留了无刷电机和电调接口，方便拓展使用。



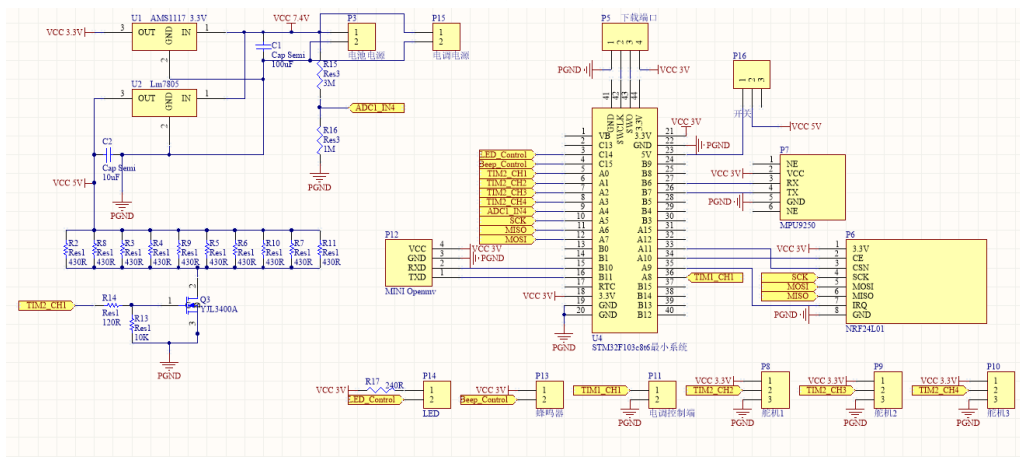
5V 版本 PCB 布线图



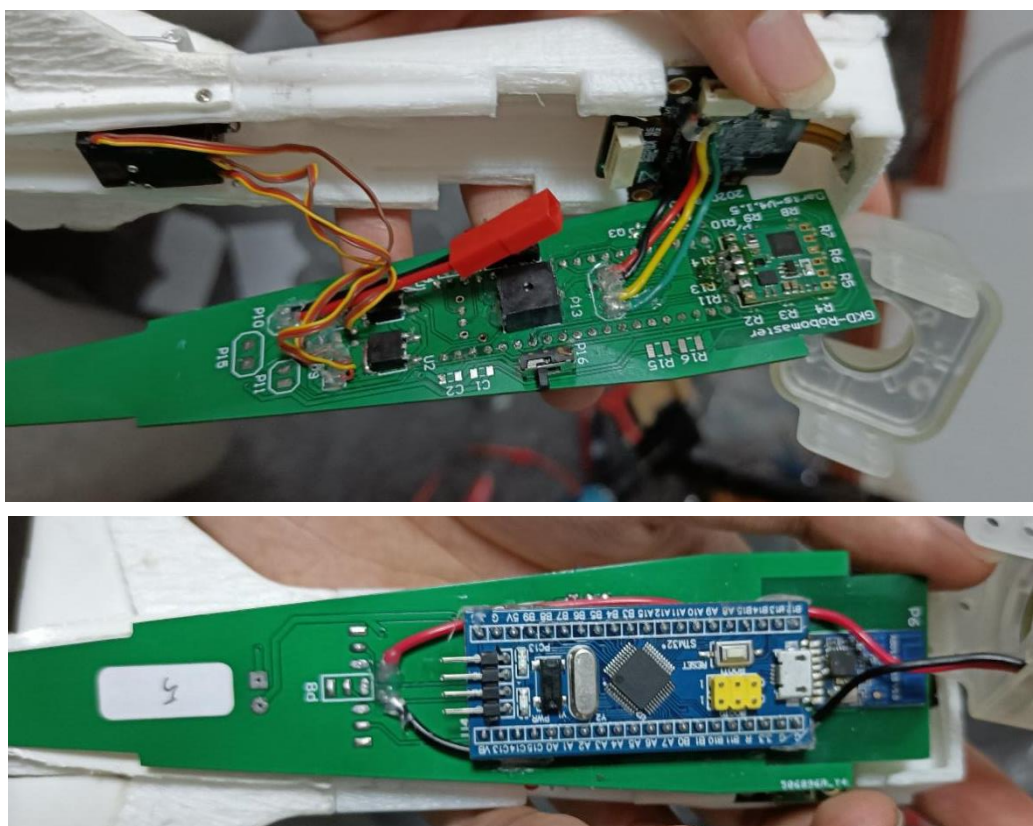
5V 版本电路原理图



3.3V 版本 PCB 布线图



3.3V 版本电路原理图



PCB 实际安装及飞镖内部走线情况

## 4. 发射架部分软件设计方案

发射架部分的程序设计中，团队主要采用的是多种电机配合硬件，按照时序来完成飞镖装填、调整发射架位置、飞镖发射等一系列操作，并且连接裁判系统来判断发射时机。

整体的发射流程为：先由直流推杆电机和底部 GM6020 电机调整发射架高度和左右偏移，随后 M3508 电机拉动滑块向下，到达底部的按动开关停止，由一个底部舵机卡住底座。随后第一枚飞镖由舵机控制装弹，同时 M3508 电机复位。此时发射已经蓄力，M3508 碰到顶部按动开关后底部舵机等待，收到发射指令后立刻发射第一枚飞镖。

第一次发射完成后，装弹 GM6020 旋转 90°，再重复上述操作，重复 3 次，完成所有操作。

上述为最简程序，可以满足在 40 秒之内完成整套系统的运行。发射架的其他功能见下。

### 4.1 调整发射架的位置

调整发射架的位置，主要是其高度和水平方向偏移，并且通过陀螺仪感知

到当前发射架发射平面当前的水平倾斜角与左右方向的“航向角”，也就是横摆角，从而实现闭环调节。

为了保证发射的稳定性，团队采用多种方式保证调节的精确性。一是直流推杆电机的同步。经测试，直接对两个直流推杆进行调节时，同步效果不理想。硬件同步轮难以设计并难以保证稳定性，经试验，增大 PWM 波形的时钟频率可以比较理想的解决问题。二是控制发射架横摆角的 GM6020 电机转动角度的精确性。经过团队试验，采用 GM6020 电机的舵机模式后，在 Assistant 软件中调节 PID 的参数可以解决该问题。三是上述方案难以解决的部分姿态调节问题。团队采用了一个六轴陀螺仪测算当前发射架整体的姿态，从而通过负反馈调节增加系统稳定性。

## 4.2 飞镖的装填

该部分的控制由以下几个部分组成：一个处于舵机模式的 GM6020 电机，六个 180°舵机（其中四个用于对飞镖抓取与释放，两个用于同步升降抓取机构来保证飞镖能够成功装填）。

GM6020 电机的精确控制依然通过调节内置的 PID 参数实现。两个同步升降的舵机通过调节 PWM 波形的时钟频率保证。

## 4.3 飞镖的发射

为飞镖提供动能的发射滑块由拉线盒连接，通过 M3508 电机实现对该滑块的拉取。M3508 电机的驱动方式为 PWM 波形驱动。通过滑轨前后安装的两个微动开关控制电机的转动方向，避免拉取长度过长或过短以损坏机构。

发射则由舵机控制的挡板实现，具体情况在硬件部分进行介绍。

## 4.4 裁判系统连接A型开发板

将裁判系统的串口与 A 型开发板的串口连接，接收裁判系统的指令。

飞镖发射系统需要的指令仅有 0x020A 一种，因此我们在程序中仅对指令进行识别，优点是简单易行，缺点是为后续拓展带来不便。

在初版程序中，程序为收到裁判系统传递“发射舱门正在开启”命令时上膛，在收到“舱门开启完毕”命令后立刻发射，但由于缺少调试机会放弃该方案（团队搭建的裁判系统环境无法发射以上指令）。第二版方案为利用操作手切换飞镖瞄准目标的按键（即 J 键）实现对上膛的控制，在本地测试取得了比较好的结果。



## 5. 飞镖部分软件设计方案

对于飞镖部分的嵌入式编程设计，我们主要使用陀螺仪感知飞镖当前的状态，通过卡尔曼滤波算法得到当前位置的准确信息，并结合了视觉识别方法得到目标前哨站或基地的位置，随后通过 PID 算法调节当前的飞机的姿态。

对于整体程序结构，我们采用了 FreeRTOS 来实现任务管理。FreeRTOS 操作系统是一个嵌入式系统使用的开源实时操作系统，FreeRTOS 支持没有限制的任务数量和没有限制的任务优先级，我们可以任意支配单片机外设的使用数量和该外设的重要程度，方便后期进行执行跟踪和堆栈溢出检测。在整个机器人系统使用过程中，根据探索需求分为多个任务，每个任务需要设置一定的优先级和堆栈资源，由操作系统来分配每个任务所需的系统资源，根据优先级进行任务调度。 [1]

在我们的系统中，主要分为以下几个任务：

### 5.1 初始化任务（“StartDefaultTask”）

初始化任务主要作用为初始化各模块参数，并检查飞镖各模块是否正常运行。包括将飞镖尾部三枚舵面初始化、初始化接收 OpenMV 与陀螺仪 MPU 的串口通道、检测 nRF 无线模块等功能。其检测方式为尝试向以上几个模块的寄存器中写入数据，如果能够写入则认为该模块没有问题，如果无法进行写入操作则令蜂鸣器长鸣报警。当检测和初始化结束后，如果没有问题则令蜂鸣器进行一长三短的四声警报，表示初始化正常。

### 5.2 MPU纠错任务（“Task\_MPUerrorhandle”）

团队在调试时发现，有时陀螺仪会产生数据发送的错误。为了避免这些错误对系统产生不良影响，团队设计了该任务。一旦发现陀螺仪数据发送出错就将该数据帧丢弃转而准备接收下一数据帧。

### 5.3 惯性导航任务（“Task\_Guandao”）

该任务主要用于将陀螺仪传输的模拟数据进行数字化，并通过调用卡尔曼滤波任务将该数据解算为当前的姿态角。

### 5.4 卡尔曼滤波任务（“Task\_Kalman1”）

卡尔曼滤波（Kalman filter）是一种高效的自回归滤波器，它能在存在诸多不确定性情况的组合信息中估计动态系统的状态，是一种强大的、通用性极强



的工具。它的优点还有内存占用较小（只需保留前一个状态）、速度快，是实时问题和嵌入式系统的理想选择。 [2]简单的说就是我们使用加速度计与角度计得到原始的模拟数据，卡尔曼滤波可以帮助我们将二者中不同原因产生的噪音滤除掉，从而融合得到当前飞镖比较准确的姿态角。卡尔曼滤波算法的推导比较复杂，在此不做详细推导。其思想主要为将多个输入的数据进行融合，并在这个过程中对这些输入中不同的高斯噪声进行滤除。

卡尔曼滤波的公式如下：

预测过程：

$$\hat{x}_k = a\hat{x}_{k-1} + bu_k$$

$$p_k = ap_{k-1}a + q$$

更新过程：

$$g_k = p_k h / (hp_k h + r)$$

$$\hat{x}_k = \hat{x}_k + g_k (z_k - h\hat{x}_k)$$

$$p_k = (1 - g_k h) p_k$$

其中， $x_k$ 为该物理量本周期的实际值， $x_{k-1}$ 为该物理量上一个周期的实际值， $\hat{x}_k$ 来表示 $x_k$ 的估计值。 $p_k$ 为本周期的预测误差。 $z_k$ 是这个物理量的测量值， $v_k$ 是测量噪声。 $bu_k$ 代表了处理噪声，这个噪声是处理模型与实际情况的差异。 $g_k$ 为卡尔曼增益，它代表了对测量值的信任程度。 $r$ 是测量噪声 $v_k$ 的平均值，测量噪声是符合高斯分布的，一般可以从传感器厂商那里获得测量噪声的均值。我们又引入了缩放系数 $h$ ，和协方差 $q$ 。

将变量扩展到多维，用向量和矩阵替换上面的变量，就可以实现多维变量的卡尔曼滤波，下面的公式适用于多维变量。

预测过程：

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_k$$

$$P_k = AP_{k-1}A^T + Q$$

更新过程：

$$G_k = P_k H^T (HP_k H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k + G_k (z_k - H\hat{x}_k)$$

$$P_k = (1 - G_k H) P_k \quad [3]$$

## 5.5 判断起飞任务（“Task\_IfLeft”）

为了节约能源，避免运算过程中产生的高温对元器件造成不良影响，在正

式比赛时，团队设定只有当飞镖起飞后才控制舵机运动。为避免在调试时飞镖意外启动，团队判断飞镖起飞的方法是判断飞镖当前的加速度是否达到预先设定的阈值。而在调试时，为了保证远程调节参数系统数据传输的完整性，团队额外增加了只有当远程调参系统发送准备起飞指令后，再通过加速判断飞镖是否起飞。

## 5.6 PID任务（“Task\_PID”）

该任务主要负责通过 PID 控制飞镖的舵面，从而实现对飞镖姿态的控制。飞镖的状态可以用俯仰角、滚转角和航向角表示，俯仰角与滚转角由飞镖尾部横向的两枚舵片控制，而航向角由飞镖尾部竖向的一枚舵片控制。

PID 控制是一种应用非常广泛的算法。从信号变换的角度而言，超前校正、滞后校正、滞后 - 超前校正可以总结为比例、积分、微分三种运算及其组合。

比例环节会成比例地反映控制系统的偏差信号  $e(t)$ ，偏差一旦产生，控制器立即产生控制作用，以减小偏差。当仅有比例控制时系统输出存在稳态误差。

积分环节指控制器的输出与输入误差信号的积分成正比关系。主要用于消除稳态误差，提高系统的无差度。积分作用的强弱取决于积分时间常数  $T_I$ ， $T_I$  越大，积分作用越弱，反之则越强。

微分环节反映偏差信号的变化趋势，并能在偏差信号变得太大之前，在系统中引入一个有效的早期修正信号，从而加快系统的动作速度，减少调节时间。在微分控制中，控制器的输出与输入误差信号的微分（即误差的变化率）成正比关系。 [4]

综上所述，其连续形式如下：

$$U(t) = k_p(err(t) + \frac{1}{T_I} \int err(t)dt + \frac{T_D d err(t)}{dt})$$

为了保证程序设计的简便性，将其离散化，我们就得到了位置式 PID：

$$u(k) = K_p e_k + K_i \sum_{i=1}^k e(i) \Delta t + K_d \frac{e(k) - e(k-1)}{\Delta t}$$

由于在系统运行的过程中，我们没有必要保留每一个调节系数，于是再次化简，得到了如下的增量式 PID 表达式：

$$\Delta u(k) = K_p(e(k) - e(k-1)) + K_i e(k) + K_d(e(k) - 2e(k-1) + e(k-2))$$

因此，团队只要通过实验调节得到参数即可以保证 PID 调节系统的准确性

团队通过调试认为，四旋翼无人机系统中常用的双环 PID 对飞镖姿态影响不大，因此为了程序的简便性，团队采用了单环 PID。

由于飞镖存在三个姿态角，因此针对每一个姿态角团队都设计了一个单独的 PID 系统进行控制。幸运的是，显然，控制滚转与俯仰角的舵面角度可以线性叠加。

在具体控制中，团队利用陀螺仪得到的飞机姿态数据控制俯仰角与滚转角，从而控制飞机水平姿态的稳定。而航向角通过 OpenMV 输入的目标坐标调节。

## 5.7 接收nRF24L01远程数据任务(“Task\_nRFSend”)

该任务主要作用为接收远程调参系统传递来的数据。

## 5.8 远程调参任务(“Task\_RecvCMD”)

该任务的主要作用为通过远程调参系统传输到飞镖系统的数据，从而实现 PID 参数的修改。

# 6. 视觉部分软件设计方案

飞镖的视觉识别通过 OpenMV 实现。OpenMV 机器视觉库使用 C 语言编写，并为开发者提供了 Python 编程接口，通过调用 Python 接口即可快速实现算法。

视觉算法首先会对 OpenMV 进行初始化设置，将 OpenMV 内置的 OV7725 摄像头的曝光时间调至 5 微秒，以屏蔽视野中较暗的干扰光，并关闭自动增益和白平衡，以确保颜色识别的准确性。

飞行过程中，OpenMV 会持续调用 Python 接口“img.find\_blobs”以寻找特定颜色的色块，并抛弃误识别的干扰像素，保留可信度最高的色块。随后，程序提取出色块的坐标，包装成规定格式的数据帧之后通过波特率为 256000 的串口发送至单片机。

find\_blobs 方法寻找色块的过程涉及一组 LAB 色彩空间的阈值参数，只有符合该阈值范围的像素才会被判定为目标。使用 OpenMV IDE 内置的阈值编辑器找到最佳阈值，并将其编写进程序。

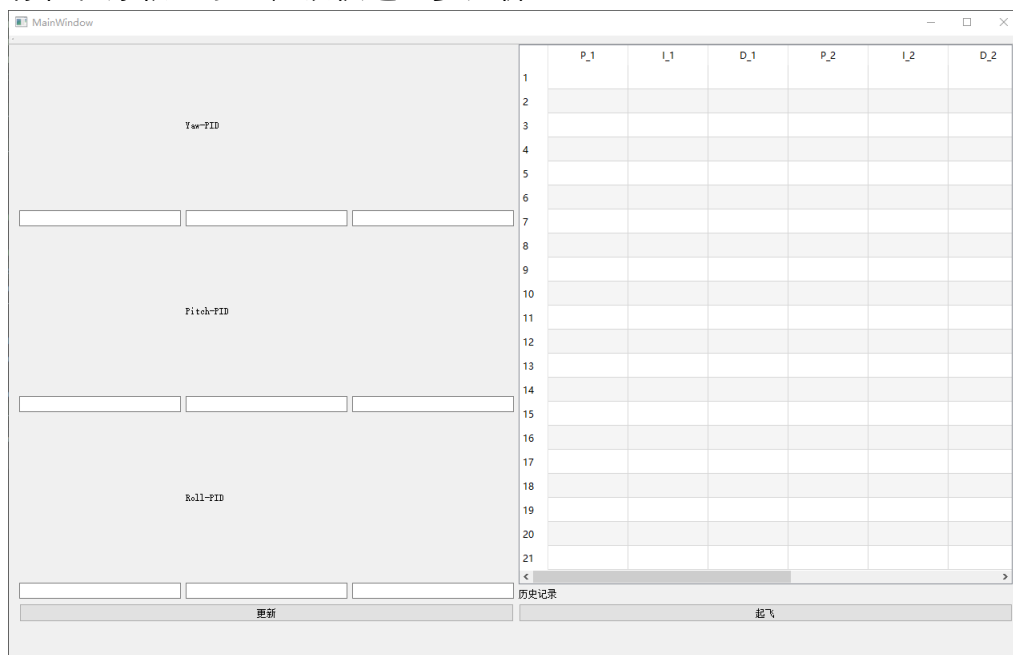
## 7. 开发工具

### 7.1 自研PID远程调参软件

飞镖的姿态控制使用了包含多个参数的 PID 算法，参数决定了控制系统中各个环节增益的大小，参数数值选择是否合理直接决定飞镖能否实现自稳和姿态调整。通常情况下，参数数值的整定需要耗费大量人力和时间，进行多次反复调整和实验，以比较不同参数的区别和优劣。而飞镖的姿态控制涉及多套 PID 参数，如果使用修改程序、重新烧写的传统方法进行 PID 参数调整，所需的时间将无法估量。

为了解决此问题，算法组和嵌入式组联合设计了基于 Python 的远程调参软件，搭配飞镖端的 nRF24L01 无线电收发芯片可实现 PID 参数的远程调整。同时，嵌入式组还使用 STM32F103C8T6 最小系统搭配 CH340 串口模块和 nRF24L01 无线电收发芯片制作了上位机端信号转发装置，用于从串口接收电脑端软件指令并通过 nRF24L01 芯片将其转化为 2.4GHz 无线电信号发送至飞镖，或从 nRF24L01 芯片接收飞镖回传的数据并通过 CH340 串口发送至电脑端软件。软件启动时会自动搜索当前可用串口并连接至转发装置。

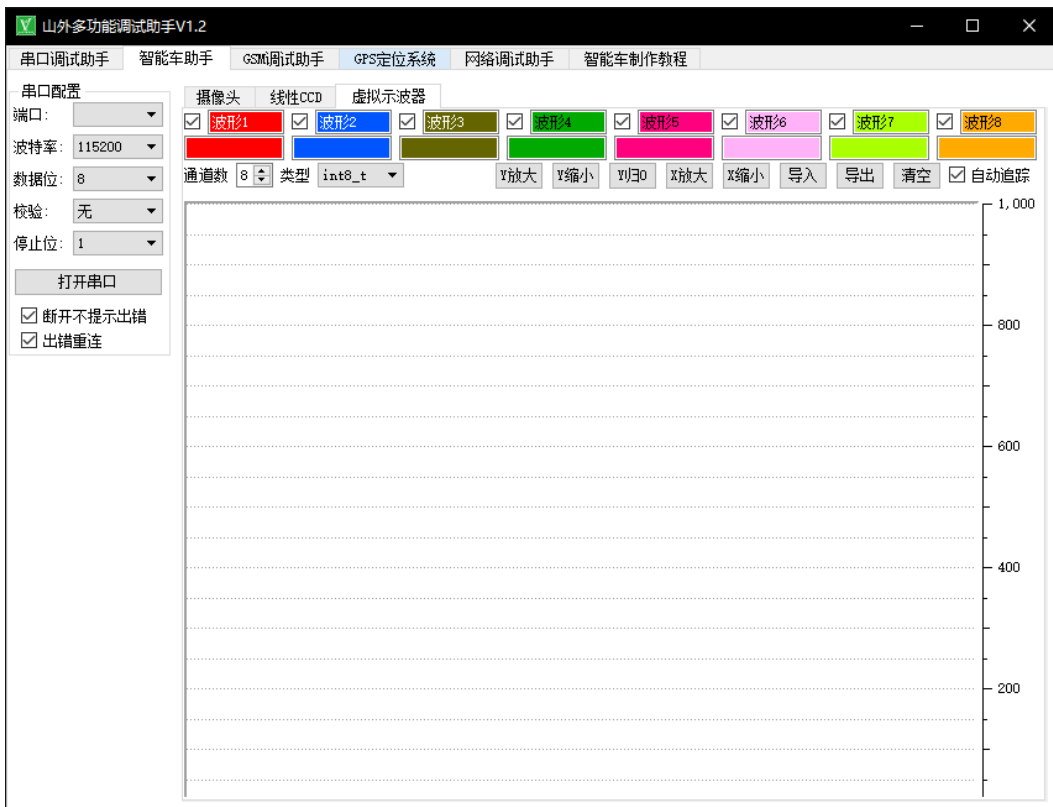
软件 UI 界面使用 PyQt5 设计，如下图所示，左侧为参数输入框，将需要修改的参数填写至此即可进行修改；右侧为参数调整记录显示，便于开发人员查看参数历史记录以选出最佳参数。同时，参数调整记录会以.txt 文件的形式保存在程序根目录下，以供进一步分析。



实际使用中，该软件搭配 nRF24L01 无线电收发芯片可实现理论上最远 10 米的远程参数调整，能够在飞镖不断电、系统不重启、发射不暂停的前提下实现参数的快速修改，节省了大量时间。

## 7.2 基于第三方软件“山外调试助手”的远程虚拟示波器

在飞镖控制参数修改后或结构变化后，需要对运行状态进行监控和分析才能找出最佳结构或参数。基于此需求，算法组和嵌入式组借助第三方软件“山外调试助手”（如下图所示）搭配飞镖 PCB 板载 nRF24L01 无线电收发芯片和信号转发装置实现了数据波形的远程显示。飞镖 RTOS 程序中加入了参数发送任务，详细可见嵌入式程序说明部分。信号转发装置只需接收无线信号，并为数据添加“山外调试助手”虚拟示波器规定的帧头和帧尾，通过串口将封装好的数据帧发送至电脑，即可在“山外调试助手”端查看波形。此外，波形还可以被保存为.csv 格式，以便通过 Excel 或 MATLAB 进行进一步的波形分析。



## 7.3 基于“飓风灯”的视觉算法参数调整和分析装置

“飓风灯”为一款能够调整亮度和色温的 RGB 灯，团队使用该 RGB 灯模拟场地中的绿灯引导标识，以调整视觉算法中颜色识别的参数。此外，算法组还通



通过对颜色进行不同幅度的改变以测试视觉算法的鲁棒性，即对场地中其他颜色光线带来的干扰的耐受程度。

## 8. 调试过程

在备赛的前期，团队进行了多型飞镖的设计和制作，并进行了发射架结构的设计。本阶段飞镖的测试过程中，如图所示，主要使用一人抛投飞镖，另一人（佩戴护目镜）手持泡沫板接取的方式进行调试，并配合前文所述的远程调参软件和远程虚拟示波器对参数进行调整，以及对飞行数据进行分析，以单独测试不同结构的性能。但由于手动投掷的重复精度有限，无法保证多次实验的初始角度和速度一致，所以只能对飞镖性能进行大致分析。在此过程中，团队对飞镖和程序进行持续的改进，明确了飞镖的技术路线。



备赛中期，初版发射架已经制作完成，已经具备了基本的装填和发射功能，但蓄力程度还无法使飞镖满足比赛要求的飞行距离。为了联合调试发射架和飞镖，如图所示，在本阶段使用发射架进行飞镖的发射，仍然安排一人（佩戴护目镜）负责手持泡沫板接取，实验主要在室内进行。本阶段的调试实现了较高

的重复精度，能够确保飞镖的每次发射都拥有相同的初速度和角度，通过调试对发射架的结构进行了大幅度的改进，并确定了飞镖的最终选型，以及优化了飞镖的控制逻辑，基本实现了姿态稳定。

本阶段的飞行测试中，针对两种主要的备选型号，机械组和嵌入式组进行了大量分析。一型飞镖的稳定性更好，空中姿态更稳定，即使在舵机不开启的情况下也有良好的飞行姿态。在受到场地气流影响时落点变化更小。但是由于主翼面积小，没有多少升力，同样初速度的情况下飞行距离相对于二型飞镖短 3 到 5 米。而二型飞镖的稳定性较差，更大的机翼让飞镖容易受到场地气流影响后发生滚转，一旦发生滚转后落点就会发生很大的变化。在使用 PID 自稳后情况有所好转，但是仍然会有滚转的情况。在综合考虑之后，认为一型飞镖是更好的选择，所以选定一型飞镖为最终结构。



备赛的后期，发射架的结构已基本确定，装弹和换弹机构也已经设计并制作完毕，本阶段主要进行了装弹换弹机构的测试和改进，以及飞镖的室外测试。调试中，发射架已经能够将飞镖发射至规定距离，团队进行了大量的实验以整定飞镖的 PID 参数，并进行了视觉算法的调试，尝试通过视觉引导飞镖打击目标。搭配前文提到的多种调试手段，获取了大量实验数据，通过对实验数据的分析，对飞镖的结构做出了多种微调。

在本阶段的调试过程中，出现了飞镖舵机接连烧毁的意外情况，在分析了

程序和 PCB 设计的多种因素后，最终将问题定位在降稳压芯片的输出电压上。舵机的额定电压为 3.3v,而降稳压芯片输出为 5v,修改电路后,问题成功解决。

此外，调试中还发现 PCB 装入飞镖机身之后，nRF24L01 无线电芯片并不能实现理论最大收发距离，推测原因为机身具有屏蔽作用，在将来的版本中可为其加入天线以提高收发距离。

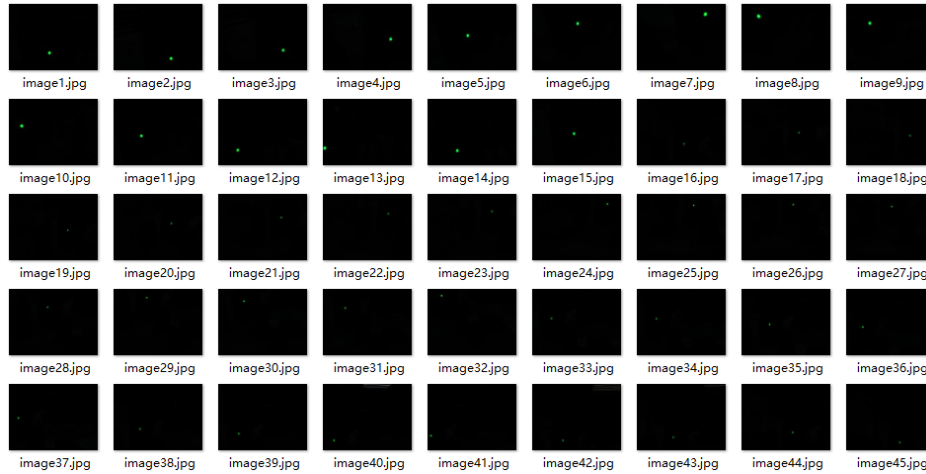


## 9. 疑难问题解决过程

### 9.1 视觉模块的选择

在进行视觉模块选型时，算法组尝试了除 OpenMV 之外的多种视觉硬件设备和算法。例如，团队尝试了树莓派 zero 搭配 OpenCV 视觉库，但实际效果证明，OpenCV 视觉库难以在低性能处理器上的 Linux 系统上获得令人满意的效果，此外树莓派 zero 因其尺寸稍大，难以安放至飞镖，于是该方案被抛弃。

团队还尝试了搭载定制 RSIC-V 神经网络加速芯片的 MAIXBit，以及自行设计并训练的卷积神经网络用于特征识别。为了更好的训练神经网络，团队自行制作并标记了引导标识数据集（如下图所示），使用 tensorflow 训练后转化为嵌入式系统专用的小体积模型，并将其部署至 MAIXBit。遗憾的是，该深度学习算法并未能取得预期效果，MAIXBit 因其性能有限，实时检测帧数较低，且只能检测低分辨率图像，无法实现预期精度。此外，识别率也未能使人满意，推测原因为实际情况下的目标检测主要识别目标的颜色而非形状，所以应将颜色识别视为首要任务。



除此之外，团队还尝试了不同型号的特定波长滤光片，试图过滤除引导标识之外的其他色光。但实际情况表明，增加滤光片后会使得摄像头只能拍摄到单一波长的色光，使不同光源失去区别，反而增加了识别难度。而且，窄带滤光片价格较高，大大增加了飞镖制作成本。于是该方案最终被抛弃，视觉最后采用了 mini OpenMV 的方案。

## 9.2 飞镖构型的迭代

飞镖的机械设计到目前为止还没有任何可以参考的开源资料，所以飞镖的所有部分均为原创设计，其中飞镖在飞镖的设计中，我们考虑到飞镖碰撞损坏的可能性非常高，所以我们始终将模块化和易维修贯彻于整个设计过程中。把所有的功能都集成在 PCB 上，避免可靠性差的接线连接，将飞镖的外壳和电路部分分离，使得不管是任何一个部分损坏都可以直接替换对应零件而不需要进行更复杂的操作。和 TPU 尾翼一体的柔性的电池舱盖，可以较为方便的更换电池。

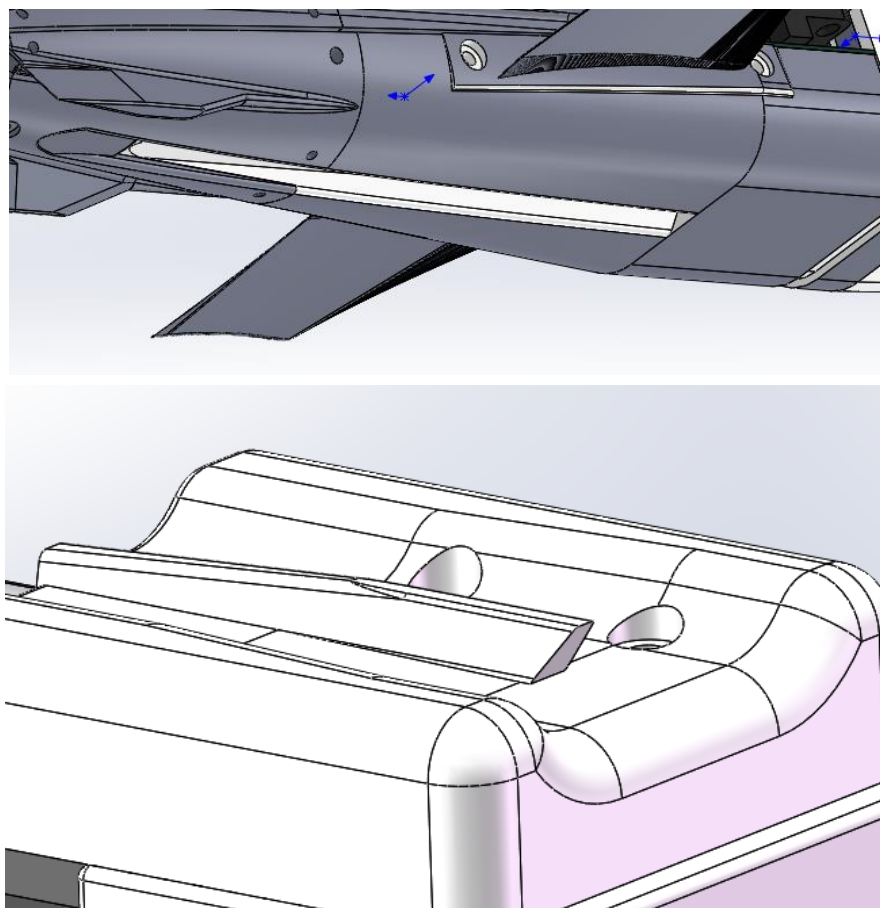
对于飞镖的气动外形，我们进行了非常多的思考和测试，从最开始的认为飞镖应该具有完全的独立飞行能力（为此设计了双翼带动力飞镖），到将飞镖看作一种受控滑翔机（为此设计了无尾三角翼，大机翼面积，大舵面的飞镖），再到将飞镖视为一种制导炮弹（为此设计了没有主翼，只有可动尾翼的飞镖），直到将两种构型结合才有了最后参加分区赛的飞镖。

飞镖的气动外形没有采用常理上应该使用的较大的机翼面积的无尾三角翼布局(事实上我们对无尾布局进行了大量实验，但是受制于场地环境和自身能力一直无法克服飞行中发生滚转的问题，无主翼的炮弹布局也存在横滚稳定性差的缺点)，而是使用了小主翼的传统布局。结合了“炮弹”和“飞机”的设计特点，既



能够和炮弹一样打出稳定的抛物线，又能够像飞机一样有着一定的滑翔能力和对 P,R,Y 三轴进行控制的能力。考虑到飞镖可能会进行频繁 PID 调参，我们设计了基于 nRF24L01 的无线调参系统，不需要进行一二次的程序烧录就能调整参数。

在前期的调试过程中，我们发现使用螺丝固定上下机身在面对剧烈冲击时会逐渐失效，造成飞镖落地后解体，最后直接放弃了一部分固定螺丝，改为前后两组扎带槽，使用扎带进行固定。取得了非常好的效果，再也没有飞镖落地后解体的情况出现。对于飞镖和发射架之间的连接方式，我们取消了早期设计中独立在飞镖底部的对位销，改为使用重新设计在飞镖底部的滑槽于发射架弹射器进行连接，反斜面设计在飞镖加速的过程中能够产生对飞镖的下压力，避免提前脱钩。对飞镖的整体气动外形影响很小，自动装填也很容易实现。



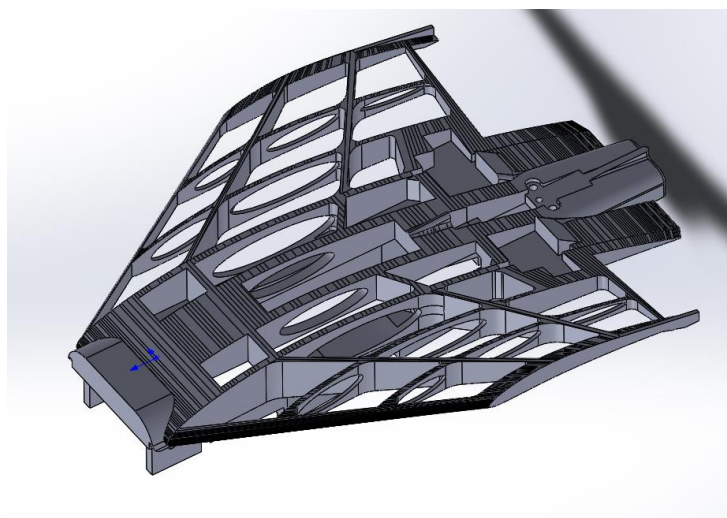
飞镖和发射架上的连接卡扣

在前期实验的过程中，我们也研发出一种适用于飞镖的复杂翼型的制作方法，厚度较厚的复杂翼型并不适合直接使用 3D 打印整体制作，会造成重量较大而且容易损坏。所以我们使用了 PLA 材料 3D 打印的镂空翼肋骨架，再使用按照图纸切割的特定形状 EPP 泡沫填充在镂空的机翼中，在稍加打磨之后使用



航模蒙皮材料对机翼进行覆膜。使用透明胶布对蒙皮的位置和边角进行固定后，使用电熨斗或者热风枪对蒙皮进行加热，使其收缩并紧绷。此时应当注意不要让热量聚集融化 PLA 骨架。在蒙皮热收缩后会自动黏附在骨架上，此时就可以揭掉用于固定的透明胶布。

使用这种方法制作的机翼相对于完全 3D 打印能够节约 50%左右的重量，并且拥有良好的表面光滑度和结构强度。在遇到剧烈冲击的工况下，骨架之间填充的 EPP 泡沫能够起到缓冲和支撑的作用。即使前缘的骨架断裂，其余部分骨架和 EPP 泡沫仍然能够维持机翼形状，其气动性能不会发生明显变化。



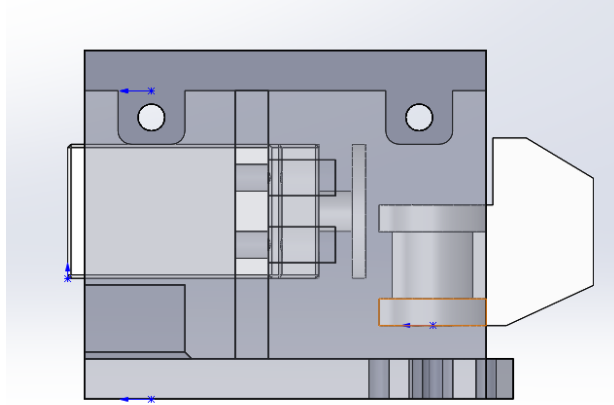
主翼 PLA 骨架示意（早期版本）

### 9.3 发射架弹鼓与释放装置的设计

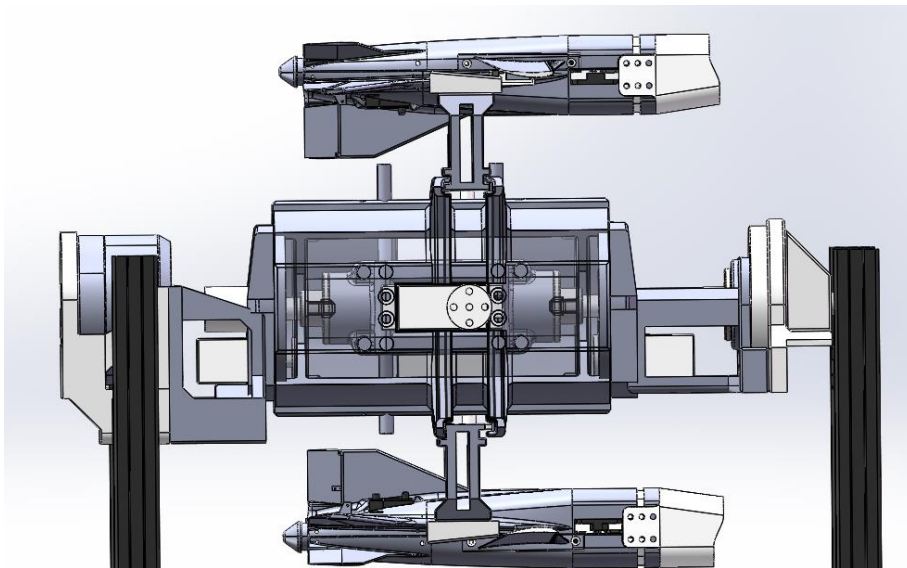
本赛季我们的飞镖发射架设计参考了西安交通大学速加网笃行战队的飞镖发射架开源 [5]，但也在一些地方进行了重新设计与改进。

释放装置：不同于普通的触发式开关，为了保证能承受橡皮筋在拉伸到最大时的拉力，我们巧妙地设计了舵机触发，利用舵机 180°的旋转范围，在舵盘上安装螺丝来按压用来阻挡滑块的打印件，使用弹簧和不锈钢杆来保证滑块的直线运动，能够承受很大的拉力并能实现自动控制释放。在橡皮筋压力过大的情况下，由于侧向力太大会导致释放装置卡死，无法释放。未来可以通过换装舵机或者使用金属零件制作舵机座以解决此问题。

但是目前在舵机可以稳定释放的情况下，飞镖初速度已经达到 16.8m/s，足够打击前哨站使用。



弹鼓装置：弹鼓采取了切分圆筒的方式形成相同的四个筒壁来分别夹取四个飞镖，每个筒壁上安装一个舵机，通过旋转使用于夹持飞镖的平行爪完成夹取和释放动作，在圆筒两侧各有一个位置调整舵机，在每个飞镖运动到位后使每块筒壁下降一定高度，使飞镖紧贴在卡扣安装座上，此时夹持飞镖的平行爪释放，飞镖在滑动一小段距离后卡在反斜面卡扣上，完成一次装填。四个筒壁通过碳管、橡皮筋和 3D 打印转接件连接在一起构成的弹鼓结构，弹鼓侧面有两块侧板，用以在其中一个筒壁下降时维持圆筒形状，提供约束，保证弹鼓正常旋转。



不足之处：（1）在长时间使用后钢丝绳会出现一定程度的磨损，较细的钢丝绳可能会在使用过程中被磨断。（2）使用的微动开关监测滑块是否装填到位的机构可能存在错误。一旦因为程序原因或者硬件问题导致 M3508 电机未能及时停转，钢丝绳会被直接拉断，这存在一定的危险性。（3）考虑到上面已经提

到的磨损问题，在能拉动滑块的情况下，也许同步带或者丝杆是更为安全可靠的选择。（4）使用 GM6020 电机加齿轮控制发射架朝向时，GM6020 电机也存在一定的虚位。导致在使用程序自动控制发射架朝向时，重复精度不高。加一套减速机构就能大大提高角度控制的精度。

## 10. 鸣谢

感谢在我们备赛过程中给予我们帮助的所有老师与同学。

感谢虽然在编外但是依然被拉过来帮我们干活的逮虾户同学。

感谢指导老师闫嘉老师，为我们提供资金支持，并为我们在寸土寸金的校内找到一间实验室。

感谢西南大学人工智能学院的相关领导，为我们提供了大量的交通和食宿报销。

感谢西安交通大学 2020 年开源的飞镖发射架设计图，为我们的设计提供了基础思路。

感谢南京信息工程大学、武汉理工大学的同学在比赛前两天与我们进行的相关技术交流，这对我们的软件设计起到了十分关键的作用。

感谢西南大学药学院团委学生会，临行前最后几天重庆有暴雨，感谢他们借给我们伞棚。

感谢竹园二舍的宿舍阿姨，我们经常从实验室晚归，免不了扰人清梦。

## 引用

- [1] 琚子晗, 白贺 和 杨喜童, “基于 Freertos 与 ARM 的智能探索机器人系统设计与实现,” *机械工程师*, pp. 37-39+42, 6 2021.
- [2] Bzarg, “How a Kalman filter works, in pictures,” 11 8 2015. [联机]. Available: <http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/>. [访问日期: 16 2 2020].
- [3] fxfreely, “滤波算法（四）——卡尔曼滤波算法,” 2 3 2020. [联机]. Available: <https://blog.csdn.net/bhniunan/article/details/104607668>. [访问日期: 5 3 2020].
- [4] tingfenghanlei, “PID 控制详解,” 16 12 2018. [联机]. Available: <https://blog.csdn.net/tingfenghanlei/article/details/85028677>. [访问日期: 15 10 2020].
- [5] shiyw50, “[开源汇总] RM2020 西安交通大学速加网笃行队飞镖发射架机械开源,” 19 9 2020. [联机]. Available: <https://bbs.robomaster.com/forum.php?mod=viewthread&tid=11041>. [访问日期: 18 12 2020].