



**GMaster**  
XJTLU × UoL

# Technical Review (Sentry)

## 技术报告 (哨兵)

### **Liverbot GMaster**

at Embedded Artificial Intelligence Hardware University-Enterprises Joint Key Laboratory

Email: [TeamGMaster@xjtlu.edu.cn](mailto:TeamGMaster@xjtlu.edu.cn)

Address: 8, Chongwen Road, SIP, Suzhou

IR724, International Research Building, South Campus, Xi'an Jiaotong-Liverpool University

# 目录

|      |                    |    |
|------|--------------------|----|
| 1.   | 机械部分 .....         | 4  |
| a)   | 总体介绍 .....         | 4  |
| i.   | 综述 .....           | 4  |
| ii.  | 赛季情况 .....         | 4  |
| iii. | 整体参数 .....         | 5  |
| b)   | 底盘设计 .....         | 6  |
| i.   | 设计思路 .....         | 6  |
| ii.  | 设计方案 .....         | 6  |
| iii. | 实测问题 .....         | 7  |
| iv.  | 改进迭代 .....         | 8  |
| c)   | 云台设计 .....         | 9  |
| i.   | 设计思路 .....         | 9  |
| ii.  | 设计方案 .....         | 9  |
| iii. | 实测问题 .....         | 11 |
| iv.  | 改进迭代 .....         | 12 |
| 2.   | 电控及硬件部分 .....      | 15 |
| a)   | 哨兵控制系统总览 .....     | 15 |
| b)   | 发射机构电源可控输出方案 ..... | 16 |
| c)   | 控制和自瞄 .....        | 17 |
| d)   | 底盘控制和碰撞检测 .....    | 18 |
| 3.   | 视觉部分 .....         | 21 |
| a)   | 综述 .....           | 21 |
| b)   | 视觉方面 .....         | 22 |
| i.   | 构架总览 .....         | 22 |

|      |                        |    |
|------|------------------------|----|
| ii.  | 帧处理算法 .....            | 23 |
| iii. | 性能测试结果 .....           | 24 |
| c)   | 通讯方面 .....             | 24 |
| i.   | 高鲁棒异步 USB 虚拟串口通讯 ..... | 24 |
| ii.  | 下位机运算压力分流 .....        | 25 |
| iii. | 黑匣子 .....              | 26 |
| iv.  | 自动裁判系统 ID 选择 .....     | 26 |
| 4.   | 参考文献 .....             | 27 |

GMaster Official Document

## 1. 机械部分

### a) 总体介绍

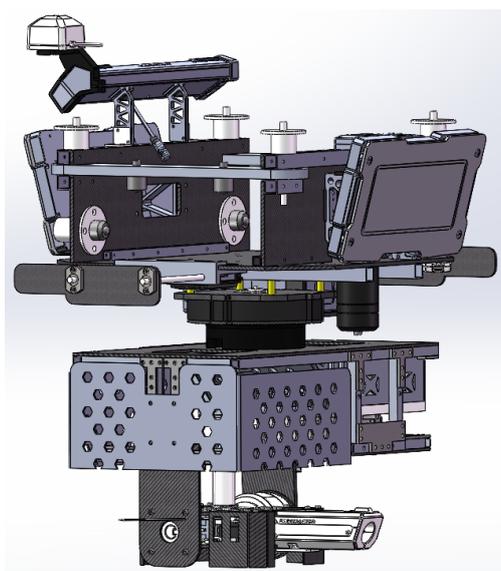
#### i. 综述

哨兵机器人得益于现代人工智能技术的支持，在 RoboMaster 机甲大师比赛实战中应用与发展。本文基于哨兵机器人的设计，调试与优化，针对哨兵机器人的机械结构与软件系统的设计，以实现比赛中全自动运行、完成智能识别击打等功能。依据实际比赛环境下外界因素对机器人实际表现的影响，优化各个功能的算法。经测试验证，哨兵机器人的稳定性和表现较上赛季有所提升，部分遗留问题得到解决，但仍存在不足和改进空间。针对实战测试发现的问题，提出下赛季改进方案。

#### ii. 赛季情况

在 2019 赛季中，哨兵机器人虽然已经出现，但是并不是所有队伍的标配。哨兵机器人具有发射 17mm 弹丸的能力，是除了 2021 赛季出现的全自动步兵外的唯一全自动兵种。本队在 2019 赛季中虽然具备哨兵机器人，但是存在造价贵、发射机构卡弹等问题。因为突如其来起来的疫情，2020 赛季线下比赛取消，队伍没有实战检测的机会。2021 赛季，在学习并结合一些哨兵机器人优秀开源资料后，确定了设计方案，主要通过机械、电控手段来解决遗留问题。

iii. 整体参数

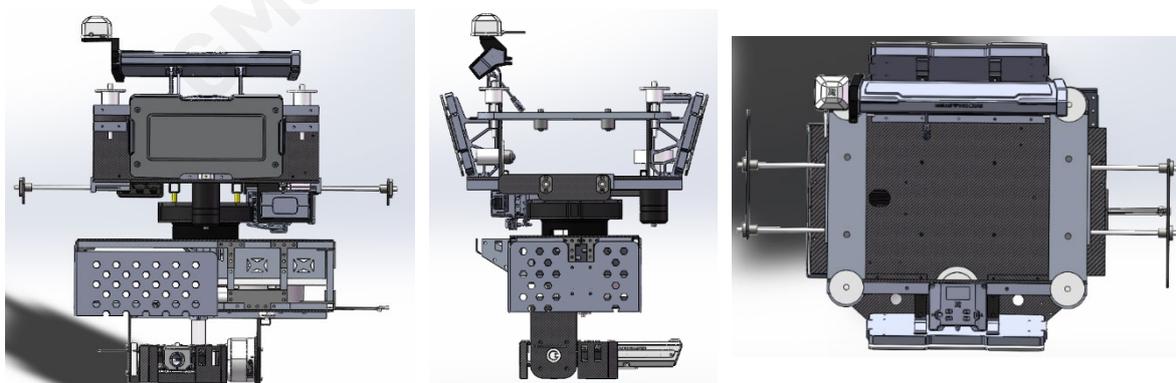


图表 1 哨兵机器人整体效果图

整车采用碳纤维板、玻璃纤维板与铝加工件结合的方式，利用切割、钣金、铸造、3D 打印等加工手段，来使整车的重量以及外观都达到预期的效果。为了便于后期的控制，采用了导电滑环实现云台的 360° 旋转。

| 参数名称        | 基本参数            | 备注             |
|-------------|-----------------|----------------|
| 机器人尺寸       | 595 × 403 × 521 | 长 × 宽 × 高 (mm) |
| 重量          | 10.6 kg         | 含裁判系统 2.63 kg  |
| 拆卸平均用时      | 30~35 秒         |                |
| PITCH 轴角度范围 | -15~30 度        |                |
| YAW 轴角度范围   | 360 度           |                |

图表 2 哨兵机器人整体参数



图表 3 哨兵机器人正、侧、俯视效果图

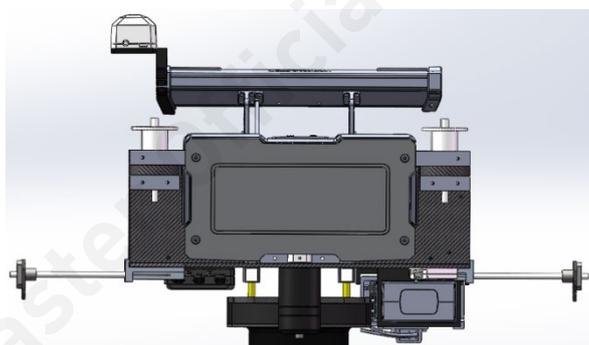
## b) 底盘设计

### i. 设计思路

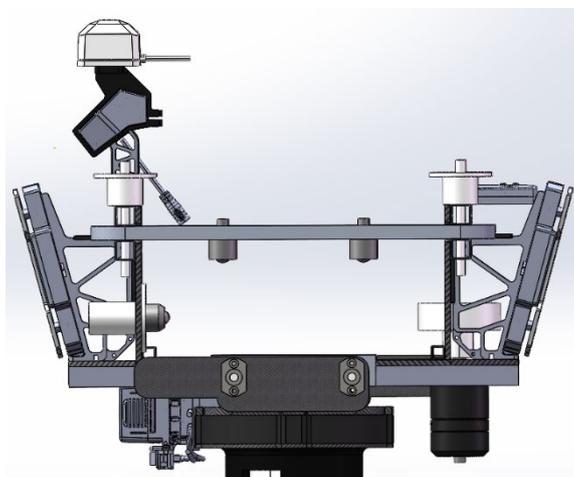
2021 赛季哨兵轨道延续 2020 赛季的直轨，哨兵运动难度不高。并且本赛季新增前哨站的机制，哨兵在战场上的位置靠后，对于小陀螺已经普遍应用的战场环境，能够快速移动躲避伤害，以保存基地护盾成为更多队伍哨兵机器人的优先项。

### ii. 设计方案

经过对开源资料的学习和研究，本队决定采用桂林电子科技大学信息科技学院 GIRT 竞技机器人战队在 2020 赛季提出的侧边轨道驱动底盘的方法：驱动电机为 RM3508 电机，驱动轮为铝合金内芯表面氧化处理聚氨酯包胶轮，邵氏硬度为 55 度，侧边驱动的优点是能减少 3508 电机的受力从而提高功率利用率。为了能够实现哨兵机器人驱动轮紧贴轨道，既不至于脱轨空转又不至于压力过大导致移动速率降低，采用了一种可以调节驱动轮和轨道松紧度的万向球与床柱连接件的配合体。组成配合件的二者均属于标准件，价格便宜，简单实用。

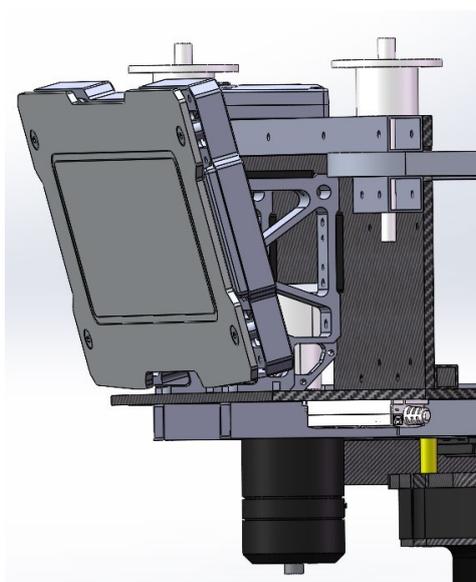


图表 5 底盘设计正视图



图表 4 底盘设计侧视图

底盘利用装甲板支撑架做为侧板加强固定件。



图表 6 装甲板支撑架加强侧板示意图

### iii. 实测问题

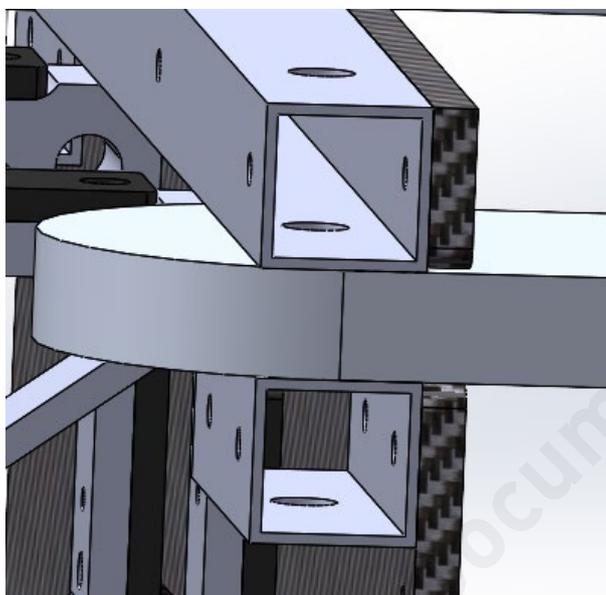
没有经过实测的设计不是完整的设计，设计很理想，但是实物装配就会发现种种问题。针对开源资料的实物装配，本队伍总结了一些问题，并且提出了改进和迭代方案（详见 [iv. 改进迭代](#)），问题如下：

1. 底盘设计没有充分考虑电控布置线路的需求，缺少更多走线孔和镂空减重设计。
2. 灯条模块一侧的快拆插销安装的时候与灯条模块干涉。



图表 7 灯条模块与快拆插销干涉示意图

3. 侧板固定使用的铝方管所用材料为 6061 铝合金。因为 6061 铝合金制成的方管为空心本身刚度不足，其挤压产生的形变随着时间的增加逐渐加剧，导致插销孔错位。因此本来应该是为了方便快速安装和拆卸底盘模块的快拆插销无法顺利插入，拆装时间增加。



图表 8 插销孔错位位置示意图

4. 侧边安装的两个万向球、上下安装的四个牛眼轮和四个万向球为从动轮，RM3508 电机驱动的聚氨酯包胶轮为主动轮。只有一个侧面主动轮而从动轮过多，哨兵机器人在轨道上所受摩擦力大，相同功率情况下在轨道上移动较为缓慢，逃逸躲避子弹能力不佳。
5. 哨兵机器人在轨道来回移动所采取的原始方案是使用红外测距，但是红外效果不佳，反馈不够及时，经常会发生与轨道立柱碰撞损坏的情况。

#### iv. 改进迭代

针对实测过程中出现的问题，本队伍提出部分解决方案，有些成功地解决了问题，有些问题仍希望在今后的赛季得到解决。

针对问题一：本队伍通过在设计部分电控与机械协作的方式，增加孔洞来方便走线；在保证机械鲁棒性的同时适当镂空减轻重量。

针对问题二：本队伍更换其他种类的快拆插销，选择两侧不同长度的插销解决干涉。

针对问题三：本队伍通过选择 6061 实心铝管替换 6061 铝方管，可以有效减小形变，但是重量的增加无法避免。希望未来可以找到更好的替换方案。

针对问题四：本队伍尝试过去掉部分从动轮，但是效果并不理想，也许替换底盘驱动轮安装方式是改进方案。

针对问题五：本队伍设计了一种碰撞机构，该机构由五个部分组成——红外模块固定座（CNC 加工）、镀铬不锈钢光轴（标准件、1 元/根）、碳纤维撞击板、光轴导套（标准件）、弹簧（标准件）。将红外模块固定座通过螺丝螺母固连在底盘碳板上；将光轴插入固定座中，通过平头螺丝将光轴 A 端固定在固定座中；将光轴导套通过螺丝螺母固连在碳纤维撞击板上；将弹簧套在光轴上；将光轴 B 端（有通孔）穿入光轴导套中，用螺丝穿过通孔利用螺母固定以保证光轴不会在哨兵机器人移动过程中脱出。

当哨兵机器人移动到轨道两端时，碳纤维撞击板碰撞轨道立柱，弹簧受力压缩形变，碳纤维板向红外模块固定座方向移动。红外模块收到信号给予电机反馈，电机反转。这样避免了红外模块受损的问题，而且可以做到快速响应。



图表 9 碰撞机构示意图

## c) 云台设计

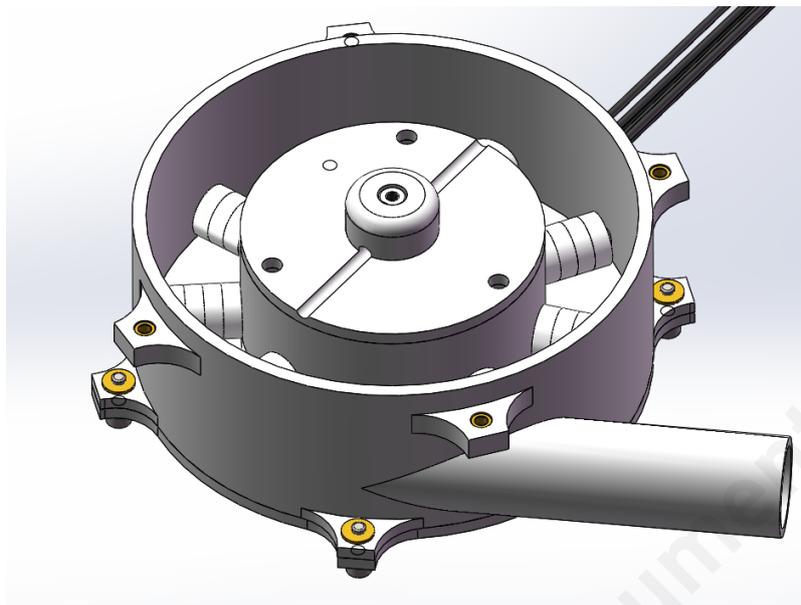
### i. 设计思路

在 2019 赛季中，本队伍设计的哨兵供弹链路长且复杂，问题排查比较困难，并且存在卡弹问题。为了解决这一问题，本队伍主要从机械、电控两方面入手。本章阐述机械部分的解决方案，包括拨弹机构、发射机构两部分。主要简化供弹链路，增加一体化程度，添加防卡弹设计。

### ii. 设计方案

经过对开源资料的学习和研究，本队决定采用 ACE 战队的圆柱轴承拨弹设计，在此基础上加以改善以适应简化后的供弹链路。设计制造实物没有使用机加工，都是 3D 打印件，结合模块化设计的思路，更换拆卸及其方便。但是在实测中仍然遇到一些问题，会在后面章节加以说明，并提出改进方案。

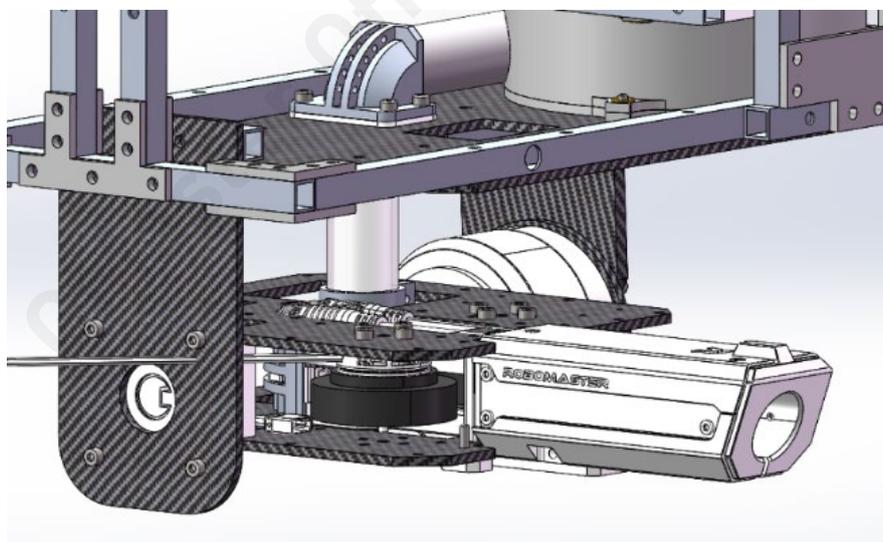
本队伍缩短拨弹轮出弹管路的长度，因为原始开源设计的长度会使 17mm 弹丸在管路中滚动时间过长还会在拨弹轮停止以后造成弹丸残留。



图表 10 ACE 拨弹机构示意图

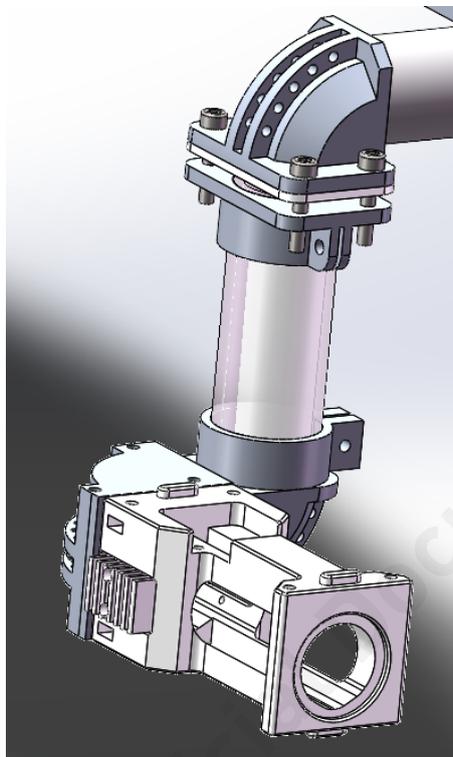
具体的内拨盘结构已经十分成熟，不再赘述。

拨弹机构将 17mm 弹丸拨出之后通过透明亚克力管进入发射机构，到达摩擦轮。结构简单，发射响应速度有所提升。



图表 11 云台发射机构示意图

本队伍考虑到成本问题以及机加工难度，供弹链路全部使用 3D 打印，并且预先使用树脂材料进行强度测试，待性能稳定后，换用韧性更高的尼龙材料打印。此举减少了装配测试过程中的不必要损耗。

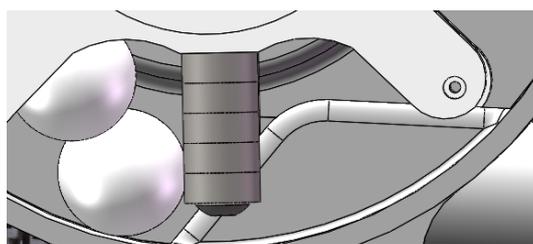


图表 12 供弹链路示意图

### iii. 实测问题

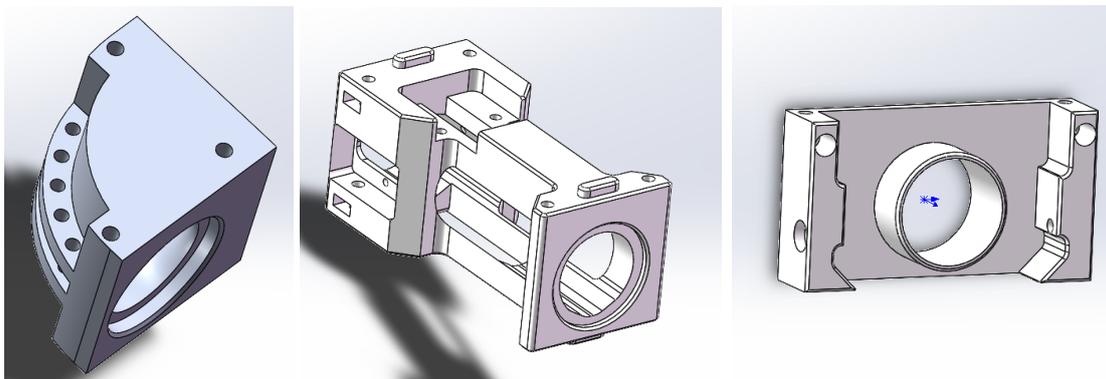
没有结合自身队伍设计特点进行修改的开源不能够直接利用到自身设计里。开源很理想，但是实物装配就会发现种种问题。针对开源资料的实物装配，本队伍总结了一些问题，并且提出了改进和迭代方案（详见 [iv. 改进迭代](#)），问题如下：

1. 拨弹仓中防止卡弹而设计的上方挡边，拨弹轮拨动弹丸对其产生碰撞。上方挡边随着碰撞的增多而断裂，碎片随着弹丸进入供弹链路造成卡弹。本队伍对其 3D 打印材料经过 PLA、树脂、尼龙三回替换，在测试过程中均发生断裂。



图表 13 上方挡边示意图

2. 在供弹过程中，因为供弹链路是分体式的，所以装配会产生误差导致链路之间不水平的情况出现，出现卡弹。



图表 14 分体 A、B、C 示意图

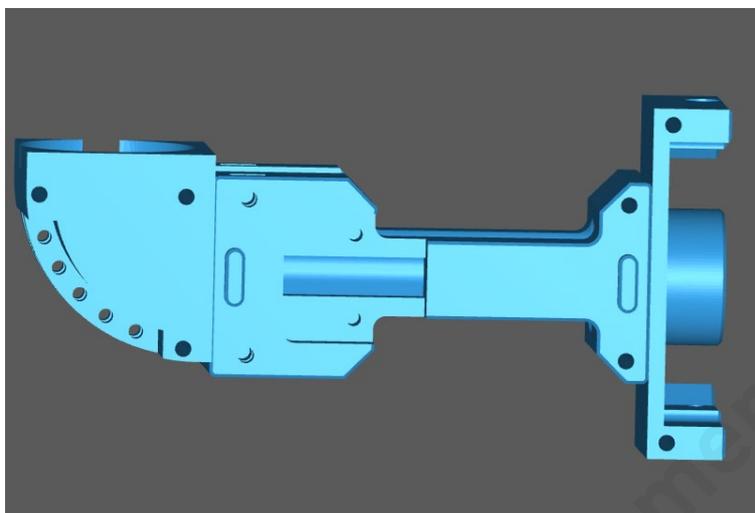
3. 因为拨弹轮拨出的弹丸通过链路后具有一定的加速度，并且哨兵机器人启动前弹路中有预制弹丸，就会出现两种状况：
  - 1) Snail 2305 摩擦轮在启动前有弹丸抵在摩擦轮上，初启动的摩擦轮转速不高，不能将弹丸擦出；后面拨弹轮持续转动送弹，弹丸无法后退，因此导致卡弹。
  - 2) 即使弹丸能够发射，因为链路上没有阻弹机构，单位时间内，拨弹轮输送弹丸的数量小于摩擦轮发射弹丸的数量。因此会出现发射与短暂空弹交替的情况。
4. 测速模块仅通过两侧螺丝固定在安装座上，因为测速模块自身重力和发射弹丸的压力，测速模块与供弹链路不水平，有一定倾斜角度导致发射出的弹丸与预期方向不一，出现散布过大情况。

#### iv. 改进迭代

针对实测过程中出现的问题，本队伍提出部分解决方案，有些成功地解决了问题，有些问题仍希望在今后的赛季得到解决。

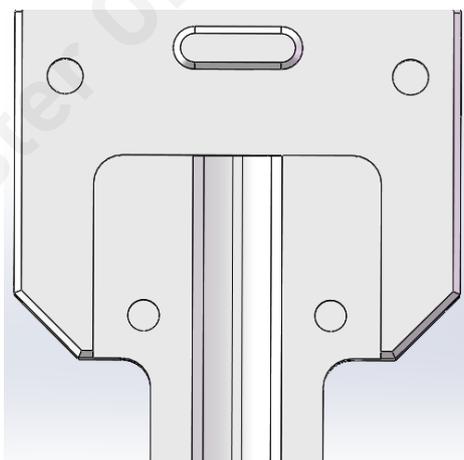
针对问题一：本队伍去掉上方挡板的设计，使用半径小的轴承和玻璃胶在原始挡板位置替换上方挡板。目的就是让轴承的弧边与运动的弹丸接触，始终有一个垂直向下的分力，使得弹丸能够紧紧贴着弹舱底部进入后面的供弹管道，能够极大减小卡弹的情况。并且因为轴承刚性强，不会出现碎裂的情况，提升了机构的鲁棒性。

针对问题二：本队伍通过在建模软件中设计修改，将分体式枪管变成一体式枪管，避免因装配出现的误差问题，极大程度地减少了卡弹风险。



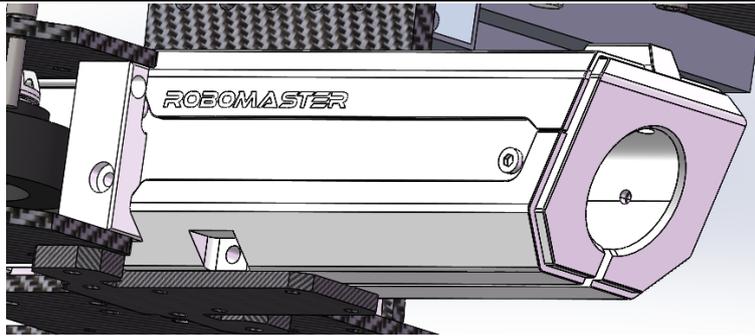
图表 15 一体式枪管 3D 打印预览图

针对问题三：本队伍在摩擦轮前方的供弹链路上方开槽，用铁丝和轴承制作一个阻弹机构（图 19 中未画出）。铁丝直径小于轴承内径。阻弹机构的作用是在拨弹轮不启动时挡住弹丸不抵达摩擦轮。随着拨弹轮的转动，后面弹丸输送挤压前方弹丸，轴承受受到向上的支持力，空间变大、弹丸通过、抵达摩擦轮。此时摩擦轮转速已经达到正常值，因此弹丸成功发射。



图表 16 阻弹机构设置位置示意图

针对问题四：本队伍在测速模块下方增加一个碳纤维托板，通过螺丝固定在云台下板和测速模块上，提供一个向上的支持力，使测速模块稳固，与供弹链路同轴心。极大程度地减少了弹丸的散布。



图表 17 碳纤维托板示意图

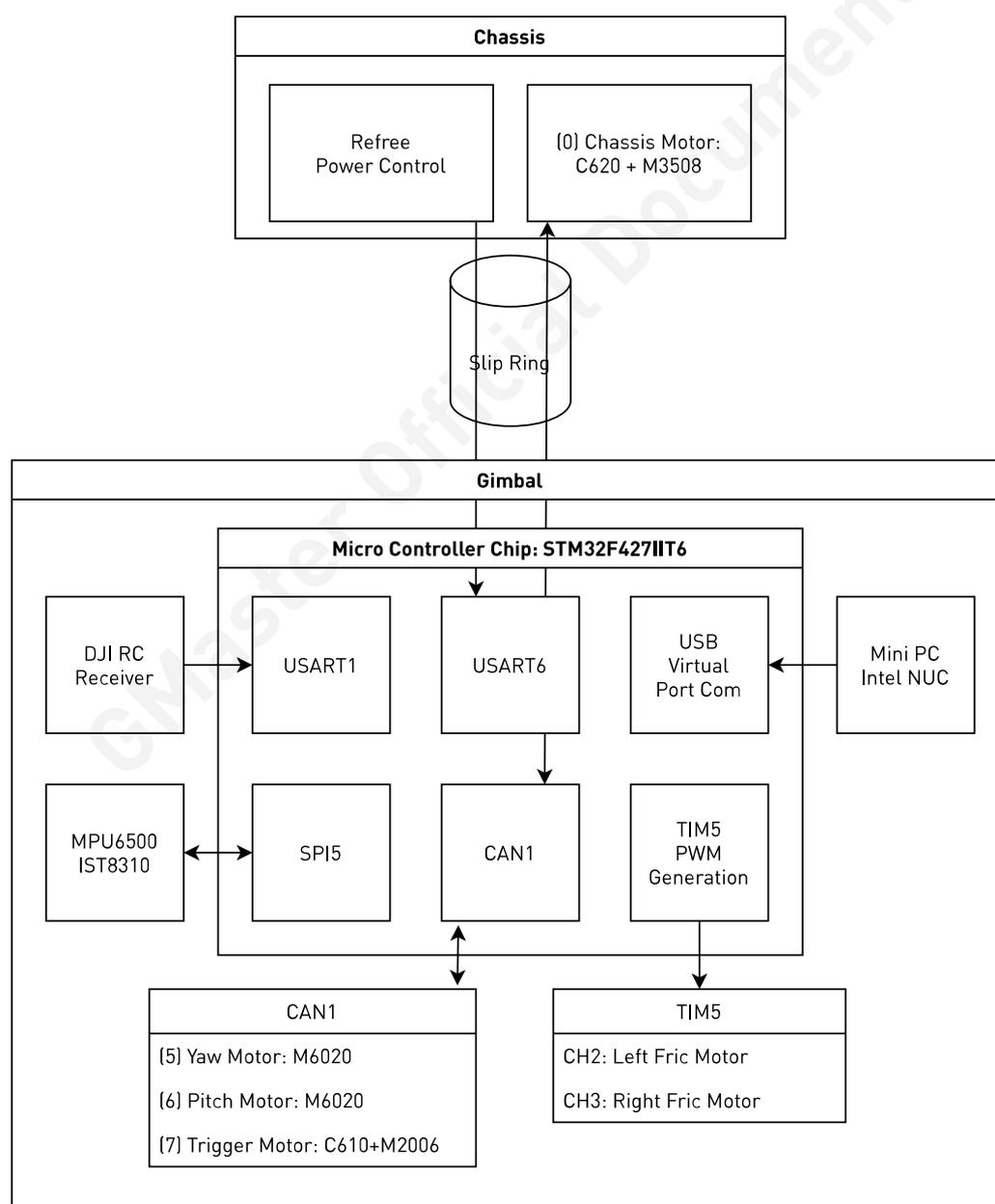
GMaster Official Document

## 2. 电控及硬件部分

### a) 哨兵控制系统总览

哨兵整体的控制系统结构中规中矩，由于没有双云台的结构，我们只需要一块开发板作为控制核心，并且放在云台上用来控制云台。我们使用了接口较为丰富的 A 型开发板。

值得注意的是在哨兵的上位机通信方面，我们抛弃了连接不够稳定的 USB 转 TTL 串口方案，而是使用了 STM32 的 USB 外设和 A 板的 USB 接口，通过 STM32CubeMX 把 USB 配置为虚拟串口功能。这样做的好处有，接线简单，连接稳定，传输速率高达 12Mbps，并且可以省去校验环节。

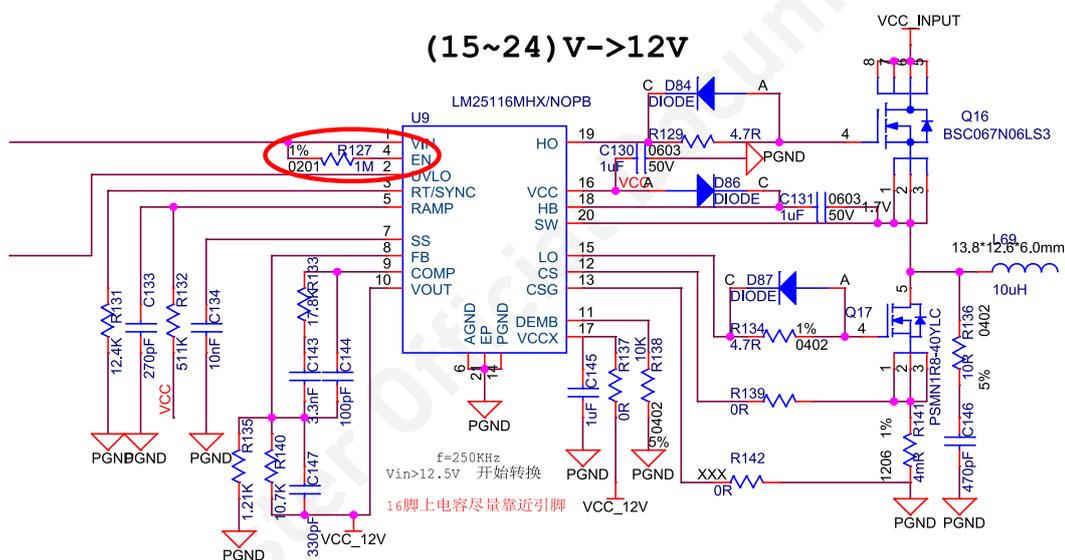


图表 18 哨兵控制系统总览

## b) 发射机构电源可控输出方案

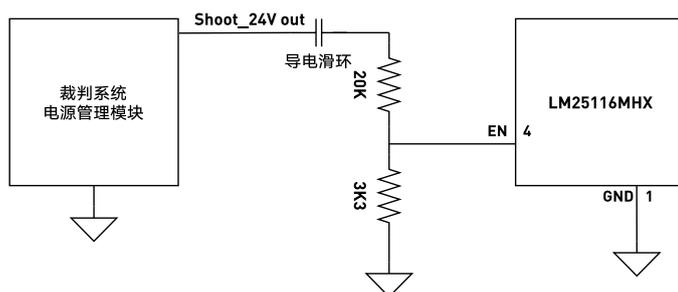
机器人制作规范和检录要求，发射机构的电源应当受到裁判系统电源管理模块的控制，也就是说当裁判系统 Shoot 接口断电，摩擦轮应立即断电停止转动。裁判系统提供的电压是 24V，而摩擦轮 Snail 电调的工作电压为 12V。常规的方案是让摩擦轮通过 24 转 12 降压模块单独连接至裁判系统取电，但这样一是会额外使用一个降压模块，二是额外占用导电滑环的有限的接口。

大疆 A 型开发板有内置的 24 转 12V 电源，我们注意到，该部分原理图中，使用的 LM25116MHX 降压芯片中，4 号引脚的名称叫“EN”，设计者在这里直接将该引脚上拉至 24V 输入，故猜测该引脚也许可以控制芯片是否输出 12V 电源。查阅数据手册证实猜想正确，该引脚电压大于 3.3V 时，该芯片才会输出 12V 供电。



图表 19 开发板 A 板降压部分原理图

于是，通过飞线，我们把 LM25116MHX 芯片的 4 号引脚飞线至一个 2.54mm 的排针处方便后续接线。之后通过下图中的分压电路来达到“通过裁判系统控制 12V 电源输出”的目的。经测试和实际检录，该方案效果良好，并且省去了额外的降压模块的空间，充分利用了 A 板的功能。



图表 20 修改后的供电方案

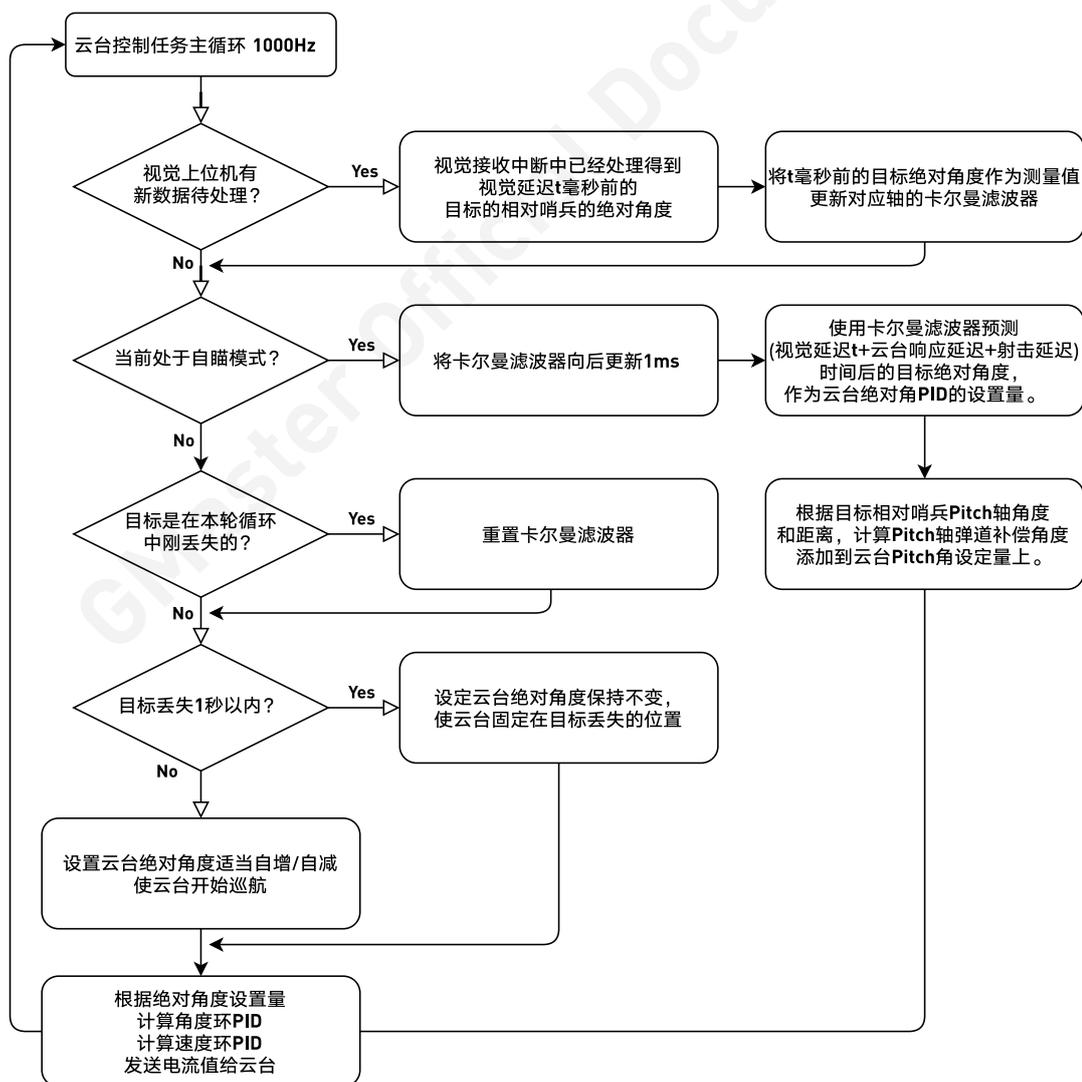
### c) 控制和自瞄

下表展示了上位机给下位机发送的数据协议及解释，使用 STM32 USB 虚拟串口通信，频率为上位机视觉程序每一次得到新结果时发送。

| 名称       | 数据类型    | 描述                       |
|----------|---------|--------------------------|
| header   | uint8_t | 数据帧头，固定为 0xF6            |
| latency  | uint8_t | 本次视觉识别的耗时（单位毫秒）典型值为 10ms |
| status   | uint8_t | 1: 找到目标; 0: 未找到目标        |
| yaw      | float   | 目标相对云台的 yaw 轴角度          |
| pitch    | float   | 目标相对云台的 pitch 轴角度        |
| distance | float   | 目标与云台旋转中心的距离             |

图表 21 上位机与下位机 USB 虚拟串口通讯

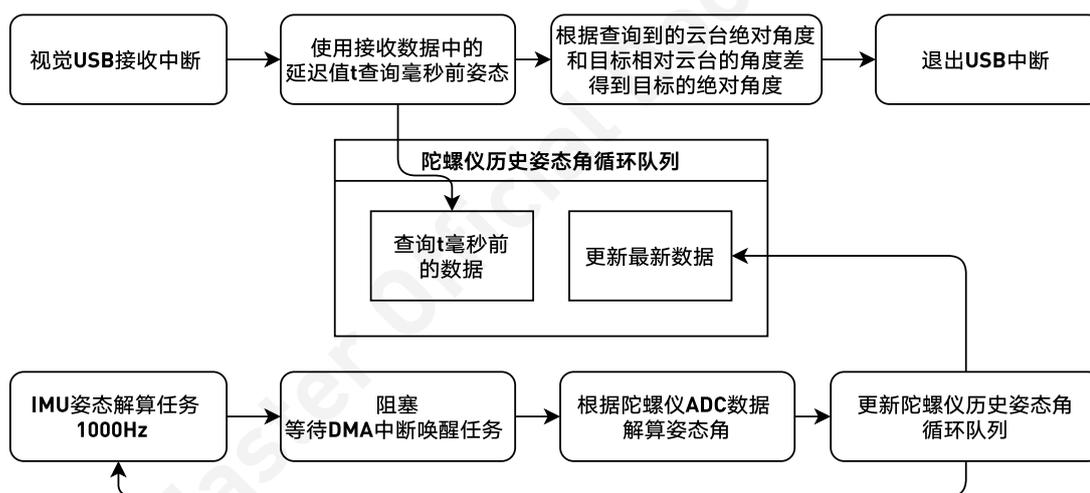
下图展示了云台整体的控制逻辑，包括哨兵与上位机的通信过程，卡尔曼滤波器的使用方式等。



图表 22 云台控制逻辑

云台任务初始化完成后，在每次循环中，主要分为三个状态来进行对应地处理：收到来自视觉的新数据，未收到新数据但仍处于自瞄模式，和目标丢失状态。收到新的数据则更新卡尔曼滤波器，处于自瞄模式时则预测一定时间后的目标绝对角度，并设定云台转动至该角度。注意在 yaw 轴和 pitch 轴的两个卡尔曼滤波器中，涉及到的角度全部为绝对角度，即目标装甲板相对于哨兵底盘的绝对角度。也就是说当底盘和目标都静止时，每次视觉中断处理得到的绝对角度不会受云台转动而变化。

由于云台经常在转动，而视觉从捕捉一帧画面到返回结果之间有数毫秒的耗时，也就是说当前得到的目标相对云台的角度，是根据数毫秒前捕捉到的画面得到的。这样如果把数毫秒前的相对角度，与当前的云台绝对角度相加，得到一个目标的绝对角度，就会受到视觉程序延迟的影响。所以我们实现了下面这样一个循环队列，包含两个方法，更新最新云台角度和查询历史云台角度。下图展示了在视觉接收中断中处理得到目标绝对角度的过程。

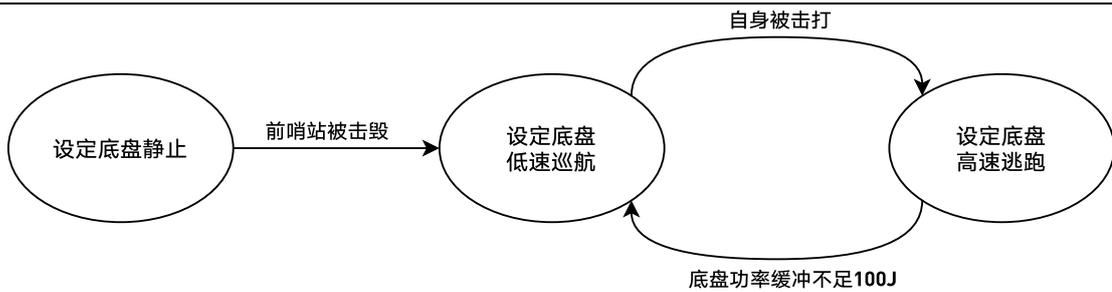


图表 23 下位机视觉数据解算过程

#### d) 底盘控制和碰撞检测

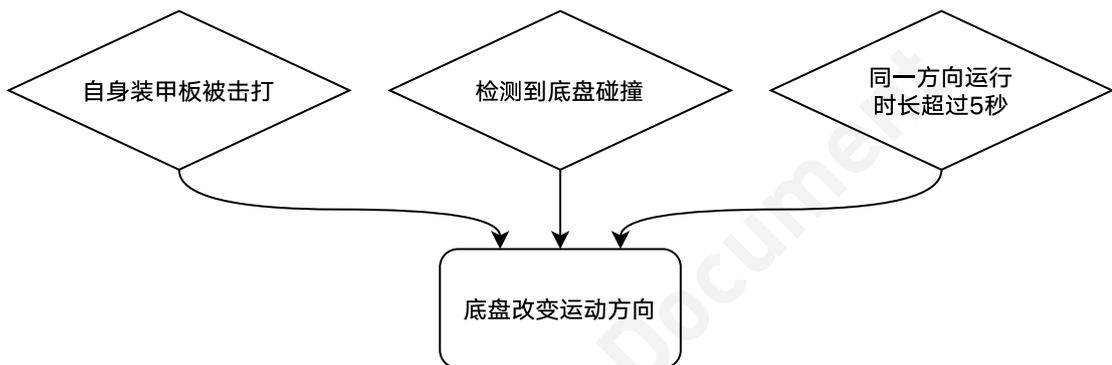
底盘控制的逻辑比较简单，分为速度控制和方向控制两部分。

下图展示了速度控制的状态机。在开发好电机驱动，并移植好功率控制算法后，需要测试出一个能够保证持久运行不会超功率的底盘设定速度。如果速度设定太大会导致一段时间后底盘功率缓冲区耗尽，哨兵无法在关键时刻逃跑甚至难以移动。



图表 24 速度控制状态机

下图展示了哨兵底盘的方向控制逻辑。满足以下三个条件任意一个时，则会调用底盘改变方向的函数来改变方向。



图表 25 方向控制逻辑

在底盘碰撞检测部分我们做了一些尝试，原本的方案是使用红外线距离传感器检测是否接近轨道尽头。但由于此类传感器返回模拟信号，波动较大，而且返回电压与距离是非线性的，并且临近比赛时机械方案的改动导致红外距离传感器的结果不一定可靠，因此最终使用了检测底盘加速度的方法。

这种方法简单来说，通过判断底盘在轨道这一个轴上的加速度是否大于某个临界值，来判断底盘是否撞击到轨道尽头。然而我们没有在底盘安装加速度计，因此决定使用云台的加速度计，然后通过云台欧拉角来变换得到底盘运动轴上的加速度。

根据我们哨兵的机械结构，从上到下分别为底盘，弹舱和云台，其中弹舱只有 yaw 轴的自由度，而云台在弹舱的基础上还有 pitch 轴的自由度。所以可以先根据云台（即开发板）的测量到的加速度，得到弹舱沿自身 X 轴和 Y 轴的加速度，再进一步得到底盘加速度。

以下伪代码（公式）根据云台加速度旋转得到弹舱（中间层）的沿弹舱 X 轴和 Y 轴的加速度：

```
accel_middle_x = accel_gimbal_x  
accel_middle_y = accel_gimbal_y * cos(pitch) - accel_gimbal_z *  
sin(pitch)
```

最后根据弹舱的 X 轴和 Y 轴的加速度，和底盘弹舱之间的相对角度，得到底盘运动方向的加速度：

$$\text{accel\_chassis} = \text{accel\_middle\_y} * \cos(\text{yaw\_ecd}) + \text{accel\_middle\_x} * \sin(\text{yaw\_ecd})$$

由于这样得到的底盘加速度会受噪音和运动等影响很大，为了提升碰撞判断的准确性，我们把上述公式得到的底盘加速度又经过一次低通滤波，通过判断滤波之后的加速度是否大于某个阈值来检测判断碰撞。这样基本可以将碰撞和别的机械扰动区分开，因为碰撞带来加速度的改变还是相对较大的。但仍然无法避免，当底盘速度较低，比如刚换向还来不及加速就碰到轨道尽头时，无法检测到碰撞的情况。

GMaster Official Document

### 3. 视觉部分

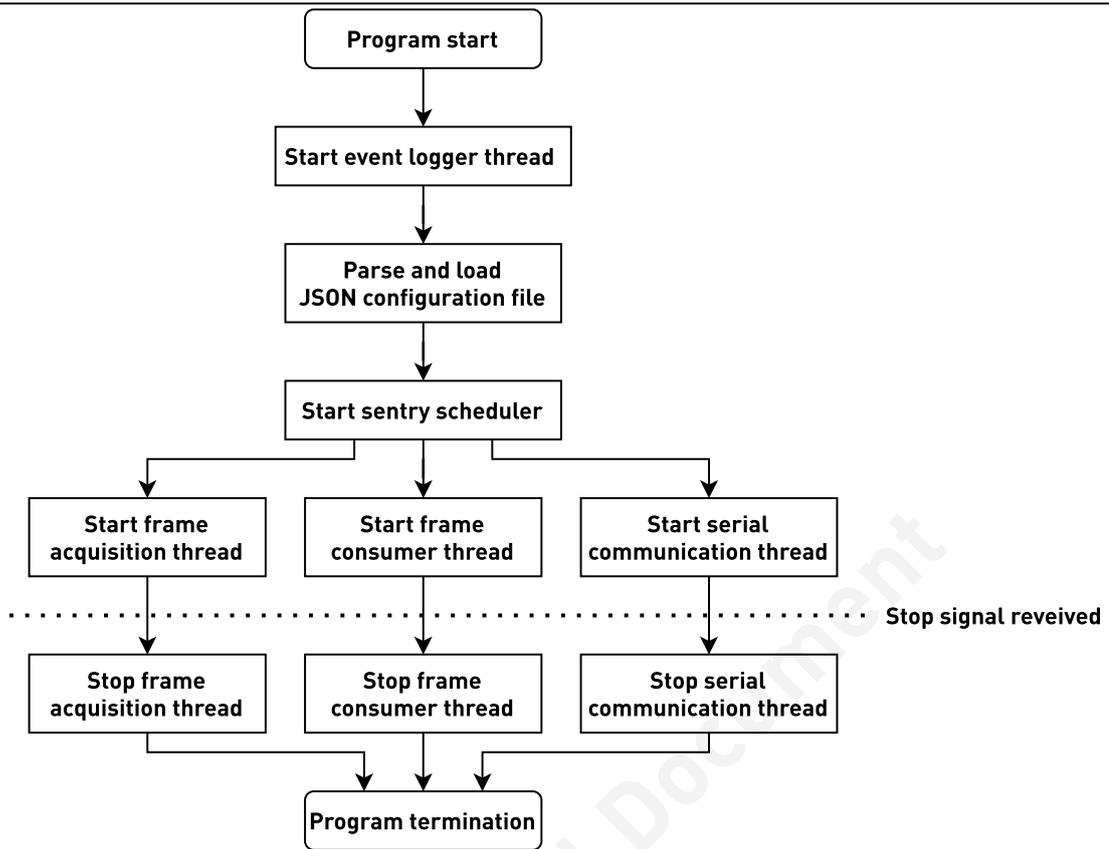
#### a) 综述

从视觉算法在 RoboMaster 2017 赛季中出现至今，许多战队都已经拥有了自己的一套成熟的视觉算法并随着比赛规则的变化不断进行迭代更新。边缘计算、神经网络等配套新型概念、算法和硬件的出现，也使各种运算的时间和空间开销相比以往降低了许多。虽然如此，许多学校机器人上所搭载的运算平台在用途上依然仅仅保持在“受限于视觉识别”这一阶段。

本战队在进行可行性讨论后，决定充分利用上位机的算力充足的优势，突破传统上位机只负责进行视觉算法的认知，扩充其作为上位机的使用目的，包括但不限于：

1. 视觉方面
  - a. 视觉识别 (基础功能)
  - b. 数据集抓取与存储
2. 通讯方面
  - a. 视觉坐标解算 (基础功能)
  - b. 下位机运算压力分流
  - c. 自动机器人 ID 选择
  - d. 机器人黑匣子
3. 日志方面
  - a. 异步程序日志记录服务
  - b. 异步黑匣子数据保存服务

以下是本赛季中哨兵机器人上位机软件工作概览：



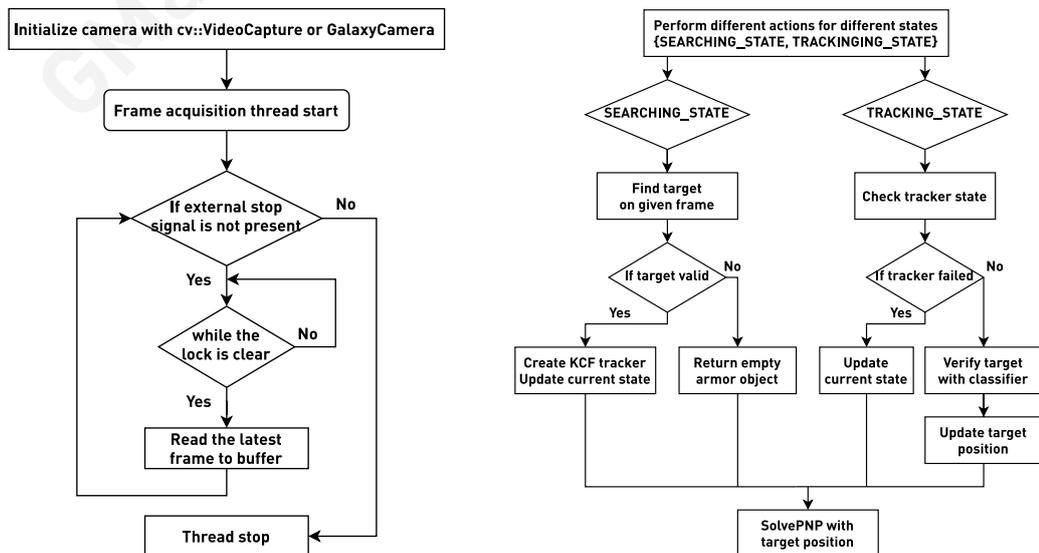
图表 26 上位机软件工作概览

下文将对本赛季上位机特色部分以及视觉识别部分分别进行介绍。

## b) 视觉方面

### i. 构架总览

在上位机的主要功能——视觉方面，主要分为两个部分：帧提供线程与帧消耗线程。以下分别是帧提供和帧消耗线程的运作构架：



图表 27 帧提供与帧消耗线程运作构架

## ii. 帧处理算法

本赛季所运用的帧处理算法大致方向为传统图像处理方法配合分类器。

传统图像处理方面主要运用为进行图像的 RGB 通道分离、二值化、开闭运算、逻辑判断和图像缩放等。

分类器方面主要运用为将传统图像处理以及逻辑判断后的可能的装甲板区域进行装甲板分类（区分内容包括颜色、装甲板数字、非装甲板目标）。分类器的实现方式为：前期使用 TensorFlow 预训练卷积神经网络；Runtime 使用 Eigen 3 导入权重进行推理。

此外，本赛季将上位机原本的“发送目标偏离画面中心像素数”的方法更新为了“发送上位机 SolvePNP 位姿解算的平移向量”。通过使用检测出的装甲板的灯条四个角点（见图表 27）进行 SolvePNP 位姿解算，可以得到相机坐标系中的  $x$ ,  $y$ ,  $z$  三个轴上的平移向量。



图表 28 所取用角点示意图

选用此方法存在以下优点：

1. 将三个轴上的平移向量发送到下位机后，方便下位机配合 IMU 实时数据实现坐标系转换（即将相机坐标系转换为世界坐标系），从而实现目标位移数据滤波和目标移动的预测。

选用此方法存在以下缺点：

1. 能进行 SolvePNP 得出相对可靠的数据的前提是摄像头已经进行了较为精细的校准。目前我们使用的校准方式是传统的棋盘格照片结合 MATLAB 内置的相机校准应用进行校准。我们得到的最优结果为重投影误差 0.08 px。

2. 根据多次实验结合 SolvePNP 本身求解原理，我们发现 SolvePNP 结果对于像素变化非常敏感，即进行 SolvePNP 解算前的输入图像分辨率越高，最终解算得出的结果会更加可靠和稳定。然而受限于算法优化能力和上位机本身算力，目前可以接受的最大分辨率（能保持稳定性能的分辨率）为  $640 \times 480$ 。这个限制条件间接导致了解算结果的不稳定性。
3. 由 2. 可知，解算对于像素变化较为敏感。为了降低输出结果的噪音，我们选择对相机进行低曝光处理。这带来的好处是更加稳定的数据，但是也对人工调教和场地灯光提出了更高的要求。  
在深圳的实际场地中，我们发现赛场上光照不充足和不稳定性对于最终的解算结果影响较大。这也是我们在下一个赛季在视觉算法部分需要着重改进的地方。

### iii. 性能测试结果

经过实验，此方法在性能上的表现良好。

具体 Benchmark 结果为：光线复杂场景下（即疑似目标较多的情况下）对于  $640 \times 480$  尺寸的输入图像处理总时长为 7~10 毫秒，最坏情况不超过 15 毫秒，满足下位机控制循环的实时性要求。

在光线充足的情况下，使用此方法配合下位机算法（移动预测与弹道补偿）可以使反小陀螺命中率达到 70% 左右（来自实验室数据）。

### c) 通讯方面

在上下位机通讯方面，在本赛季我们在上位机引入了以下机制：

#### i. 高鲁棒异步 USB 虚拟串口通讯

在单片机控制程序研发过程中，我们发现 STM32 微控制器自身提供的 USB Device 特性可以被配置为 Virtual ComPort，即 USB 虚拟串口。在经过资料查找和自身反复实验后发现，使用 USB 虚拟串口通讯相较于一般外置独立 USB-TTL 设备具有以下优点：

1. USB 虚拟串口通讯可以提供更高的 Full Duplex 传输带宽（根据 ST 官方文档，带宽大约是 12Mbps）。在下文的黑匣子特性中，此优势可以在下文得到体现。
2. USB 虚拟串口的物理传输层为 USB 协议，因此在数据完整性和校验方面提供了更高的稳定性和优势。借助这一优势，我们可以直接在两

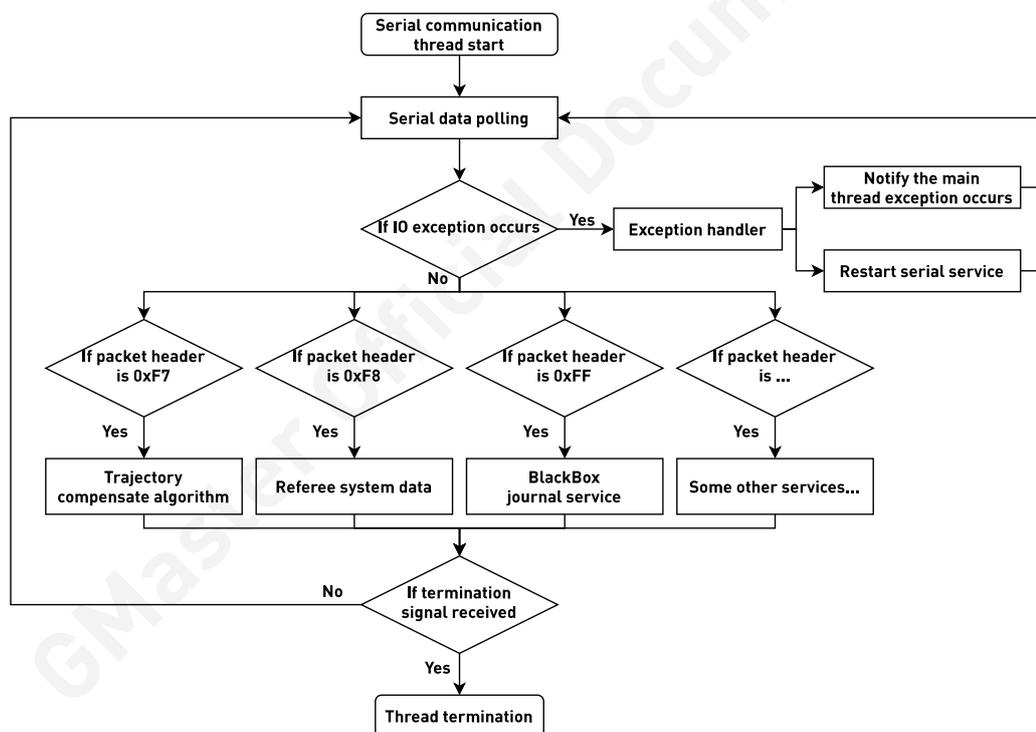
端传输数据而无需添加额外的 CRC 校验，从某种程度上来说减轻了  
下位机的计算压力（在运算压力大的情况下）。

3. 将下位机配置为 USB 虚拟串口后，可以直接在上位机中被枚举为串  
口设备（以 Linux 为例，会被内核识别为“/dev/ttyACMX”，其中 X  
为设备数）。因此不需要实现额外的驱动即可实现串口通讯。

在高鲁棒性部分，我们通过结合上位机的串口库，实现了上位机程序运行中  
下位机热插拔功能。其主要目的是为了防止由上位机或下位机意外断电或者  
接口松动而导致的信号不稳定、程序终止或崩溃问题。

在通讯协议方面，我们也实现了类似 TCP 的握手/重传以及包头区分等增强  
功能性和鲁棒性的特性。

以下是高鲁棒异步串口的工作流程：



图表 29 高鲁棒异步串口工作流程图

## ii. 下位机运算压力分流

选择 SolvePNP 后为我们提供了很多的其他可能性，其中之一就是弹道补偿  
运算。目前我们选用的算法是迭代求解，然而经过测试后我们发现将这部分  
运算放在算力受限的下位机上是不合适的，因此我们选择了通过 USB 虚拟  
串口将数据传输到上位机上进行运算的方法。

由于此算法需要结合 IMU 数据，运算的结果又被下位机的控制循环所需，因此对于实时性要求较高。我们在进行实际测试后发现，使用 GCC 编译器 O3 优化下的迭代算法可以较为稳定的在 500 微秒以内算出弹道补偿结果，并通过串口将结果传输至下位机，总消耗时长可以控制在 1000 微秒（1 毫秒）内。

这是我们对于上下位机运算压力分流的第一次尝试，也为今后更多功能的开发积累了经验。

### iii. 黑匣子

由于使用了 USB 虚拟串口，我们利用其高速与无需进行数据校验的优势实现了“黑匣子”的功能。其本质是下位机通过串口将各个电机、IMU 等的实时数据传输至上位机，并由上位机进行反序列化和序列化后保存至上位机存储。

黑匣子如其名，保存了机器人在运作过程中的各项数据，方便后期在开发调试过程中进行读取和分析以及赛后复盘。

### iv. 自动裁判系统 ID 选择

本功能基于裁判系统读取机器人 ID 数据。下位机读取数据后将本机 ID 发送至上位机，然后上位机通过 ID 信息选择敌方颜色以及开启或关闭某些兵种特有的特性。

#### 4. 参考文献

RM2020 桂林电子科技大学信息科技学院 GIRT 战队哨兵机器人开源

<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=11040&fromuid=48091> (出处: RoboMaster 论坛)

RM2020 东莞理工学院 ACE 机械 模块化拨弹开源

<https://bbs.robomaster.com/forum.php?mod=viewthread&tid=11009&fromuid=48091> (出处: RoboMaster 论坛)

GMaster Official Document