

RM 冬令营技术报告

1、项目规划

1.1 项目规划

初步计划：7 号上午前完成基本搭建，7 号下午和 8 号进行各方面调试以及软件测试。算法开始准备华容道，嵌入式配合机械完成测试。

1.2 主要设计思路

设计亮点：

- 1、上板使用碳纤维板而下板玻璃纤维板，有一定的减震功能
- 2、车型设计尽量小巧，方便上下坡，以及灵活移动。
- 3、装甲板的位置设计相对较低，延长瞄准时间

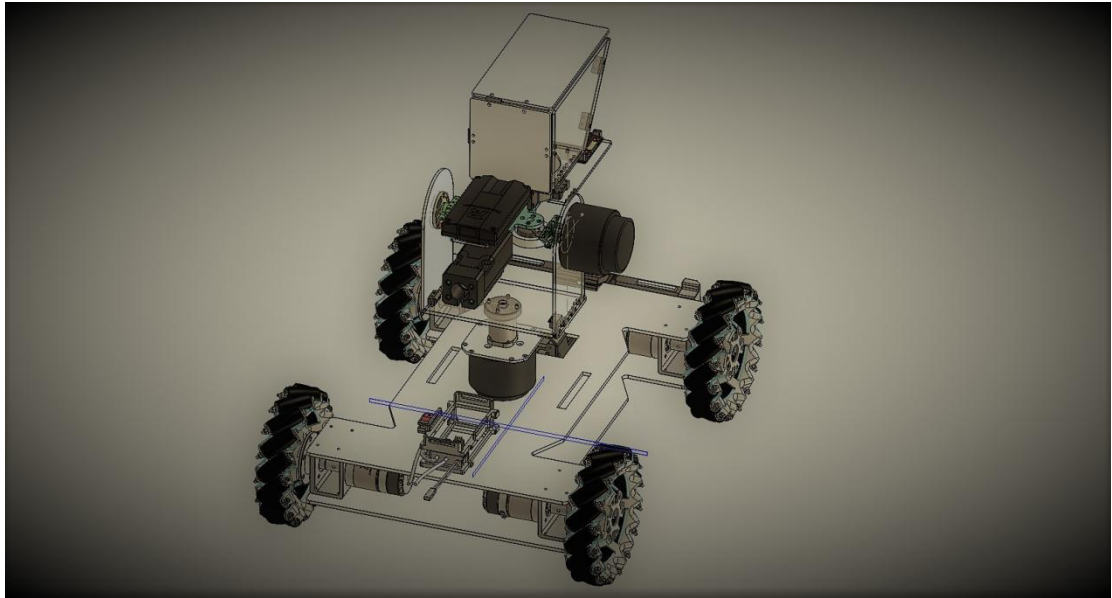
技术亮点：

1、机械臂设计（被淘汰）两个伺服通过伺服支架固定在一个伺服槽内，伺服槽一侧的挡板预留孔位可以与电机轴连接的法兰盘连接，以这个电机提供的动力来调整伺服槽的仰角，电机通过电机架固定在底盘上，另一侧的挡板预留可以安装轴承的孔位，一根轴穿过轴承将伺服槽架在一个支架上，支架固定在底盘。

（淘汰原因：太过庞大、底盘空间不足以容纳）

2、机械设计

2.1 项目模型



2.2 各模块设计思路

底盘结构：

底板：双层板结构、上板使用碳纤维板而下板玻璃纤维板，有一定的减震功能，上层板预留出走线和云台以及装甲板安装的孔位。底板由外到内逐渐缩小，减小被架空的可能性。

云台结构：

底座：在上板上预留云台底座的空位，云台采用半嵌入式结构，增强云台的稳定性。

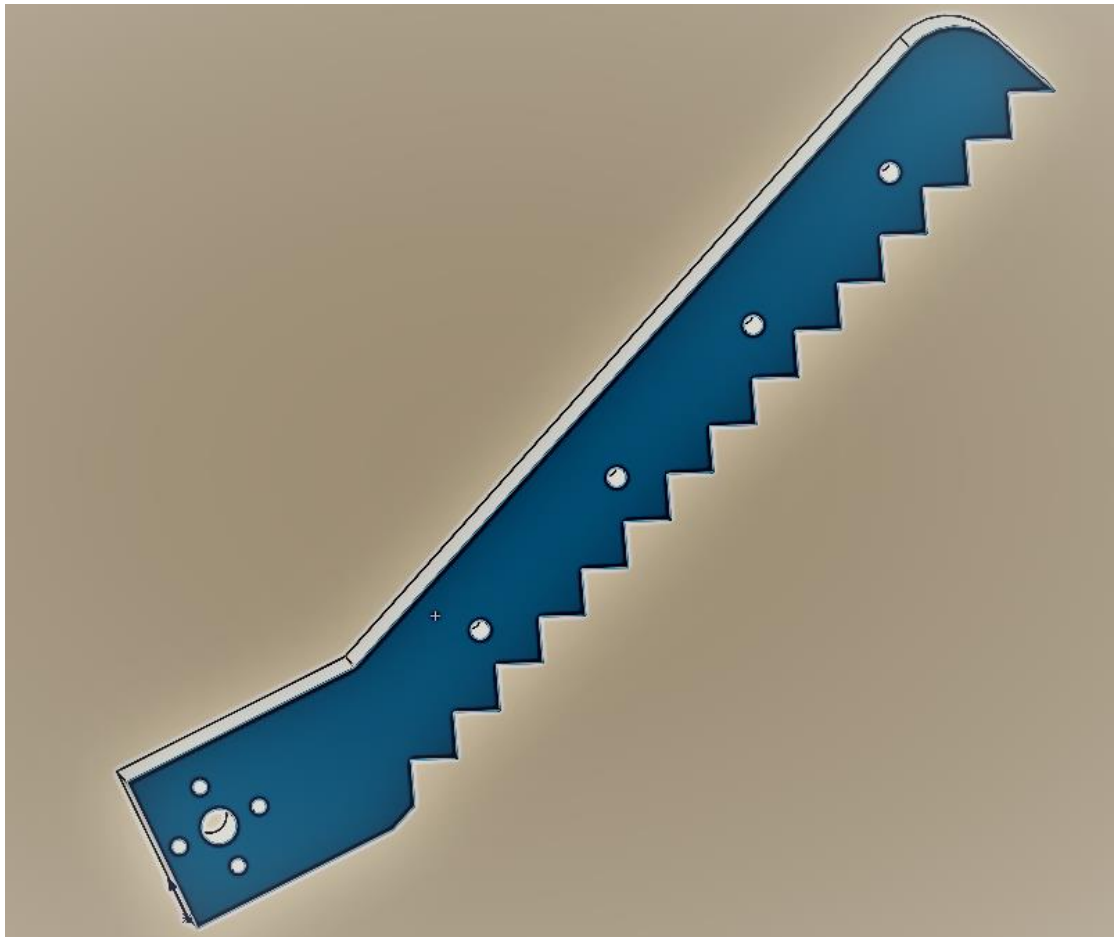
火炮：按照图纸制作，尾板加装配重，减轻云台电机压力。

弹仓：采用亚克力板结构，翻盖式顶盖，可储存 150-270 发子弹

夹取结构：

动力选型：总共有四个伺服，左右两个夹子单独驱动（控制夹子闭合一个伺服抬升一个伺服）

结构：两片此机构用铜柱连接，制成一侧夹子。



2.3 优化和改动

放弃了机械爪（因为时间问题）

原有的单板设计因为考虑云台的放置位置，所以改为上下板设计。

把裁判系统的灯柱倒放，可造成一定的视觉干扰，同时放低重心，降低翻车的可能性。

2.4 可行性测试

底盘可行性测试

场地测试：上下坡是否会翻车；下坡时云台是否会卡住，当车辆底盘被架空时是否可以自主脱离。

云台可行性测试

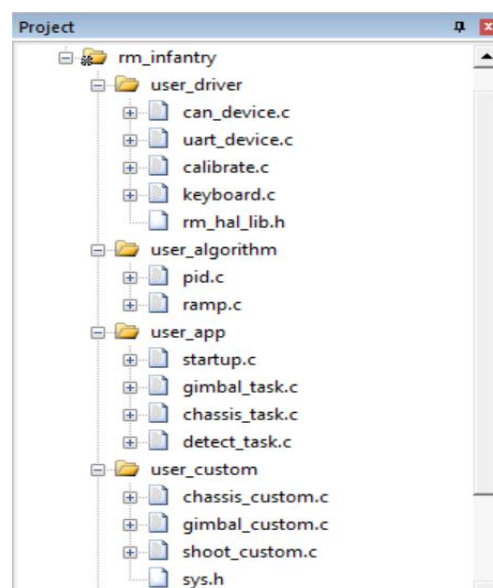
部分测试：云台是否会卡弹，射弹精确度，稳定性测试，是否会在行进中抖动过于剧烈。

3、控制部分

3.1 项目软件架构

3.1.1 软件框图

程序软件框图，工程的整体执行框架。让我们能够直观的了解项目代码的内部结构。



主要程序框架是 RM 官方给的例程，改动的地方有射频射速与云台 PID 控制等。

3.1.2 操作说明

项目控制方法介绍

3.2 云台模块

代码图：

```
void gimbal_init_param(void)
{
    //电调 bug
    osDelay(3000);

    /* 云台pitch轴电机PID参数初始化 */
    pid_init(&pid_pit, 3000, 0,
             60, 0, 0);
    pid_init(&pid_pit_speed, 5000, 2000,
             15, 0, 0);

    /* 云台yaw轴电机PID参数初始化 */
    pid_init(&pid_yaw, 2000, 0,
             25, 0.05, 0); //
    pid_init(&pid_yaw_speed, 5000, 800,
             20, 0, 0);

    /* 拨弹电机PID参数初始化 */
    pid_init(&pid_trigger, 4000, 2000,
             0.15f, 0, 0);
    pid_init(&pid_trigger_speed, 8000, 4000,
             1.5, 0.05, 0);
```

设计方案：

云台模块主要调整了 PID 的参数。从直接测试的状况来看首先 pitch 轴对于突然转动的变化反映不太灵敏，决定调高 kP 值，使其反应更为迅速。其次，yaw 轴电机在校准后仍存在不断大幅振荡的情况，判断为 kI 参数过大导致的异常情况，在调小 kI 后振荡的情况消失。

问题：

云台模块介绍，需要介绍此部分代码的逻辑框图和在调试过程中遇到的问题及解决思路，同时注明负责人。这部分需要列出项目细节

和构建思路。优质的想法并且落地实现是非常宝贵的。

3.3 底盘模块

底盘模块使用源代码。

```
46
47  /* 下面为底盘功能任务的函数 */
48  void chassis_task(const void* argu)
49  {
50      //初始化底盘控制PID参数
51      chassis_pid_param_init();
52
53      //底盘控制任务循环
54      uint32_t chassis_wake_time = osKernelSysTick();
55      while (1)
56      {
57          //切换底盘状态
58          get_chassis_mode();
59
60          switch (chassis.mode)
61          {
62              //底盘跟随云台模式，右侧拨杆在上面
63              case CHASSIS_FOLLOW_GIMBAL:
64              {
65                  chassis_control_information_get();
66
67                  //底盘跟随云台旋转控制，覆盖前面计算出来的值
68                  if ((gim.ctrl_mode == GIMBAL_CLOSE_LOOP_ZGYRO)
69                      || ((gim.ctrl_mode == GIMBAL_NO_ACTION) && (gim.n
70                      chassis.vw = pid_calc(&pid_chassis_angle, yaw_re.
```

3.4 发射模块

射速特点：为防止卡弹射不出去弹丸以及射速超裁判系统限制，我们调整至最佳射速。

```
83  /*****发射速度设置*****/
84  #define SHOT_FRIC_WHEEL_SPEED    1300 //最大为2500
85
86
87  /*****发射频率设置*****/
88  #define TRIGGER_MOTOR_SPEED      -1250 //
89
```

4、总结

4.1 项目收获

通过这次项目，我们出现了许多意外，也明白了许多的内容。

首先在比赛期间，第一阶段的布线略显混乱，在设计方面没有认真评估，导致后期返工较多，增加了许多工作量，所幸在比赛期间我们在战车上没有出现大的问题。但是我们缺乏对队友的充分了解，没有合理地安排每个人的工作，比如算法会相对清闲但是机械忙的不开交，还是有些分工上的失误，我们太看重经验了，在操作上选择了有经验的杨林千但是却没有考虑到像反应度、应变能力等问题。

学会了许多机床上的应用和操作，也从助教口中学会了很多安装顺序和用有限的材料完成任务的方法，尽管我们第一阶段的比赛输了，但是我们学会的经验不是输赢可以带代表的，希望在第二阶段我们可以有一个更好的成绩。

4.2 马后炮

对抗赛中间出现了很多问题：

- 1、在将机械爪的舵机连接在战车上之后因为主控板曾经被烧过，供电跟不上导致摩擦轮始终有一边无法转动，我们在比赛前才在一骏等助教的帮助下我们匆忙调试完成。

- 2、我们对操作手没有充分的了解，只关注了经验但是却没有熟悉考察，导致我们的操作手并不太合适，这是我们出现的失误。

3、但是我们队的战车本身我觉得是非常棒的，这来源于机械手丰富的经验和助教的指导，为我们排除了很多不合理的方案，可以及时筛选出合适的方案进行制作。