

RoboMaster 冬令营技术报告（对抗赛）

1、项目规划

1.1 项目规划

Day1: 了解熟悉队友、机器，确定主要任务，了解基本知识如矩阵、运动学分析。人员分配如下：

黄秋灿 机动应变

付郑语 嵌入式

梁宇辰 机械

刘臣轩 算法

蔡佳泽 机械

陈子讯 算法

陈永豪 后勤

Day2:

搭建底盘（优先度：最高，主要风险在于加工排队等候时间）

制作评审 ppt（优先度：高，主要风险：无）

装配官方云台（优先度：低，主要风险：无）

任务完成情况

完成三个任务

Day4:

修改底盘麦克纳姆轮装配方式、搭建上层、枪管（优先度：最高，风险：无）

设计搭建机械臂（优先度：高，风险：多次测试较浪费时间，制作时间可能来不及）

实现第一版视觉识别（优先度：中，风险：环境搭建不成功，注：若今日无法实现，则拆分一人编写、测试第二轮算法）

测试连连看算法（优先度：中，风险：无）

任务完成情况：

机械臂设计搭建未完成，其余任务完成。

Day5:

装配弹仓（优先度：高，风险：无）

搭建机械臂（优先度：高，风险：多次测试较浪费时间，制作时间可能来不及）

任务完成情况：

机械臂仍未完成，弹仓完成。

Day6: 比赛

1.2 主要设计思路

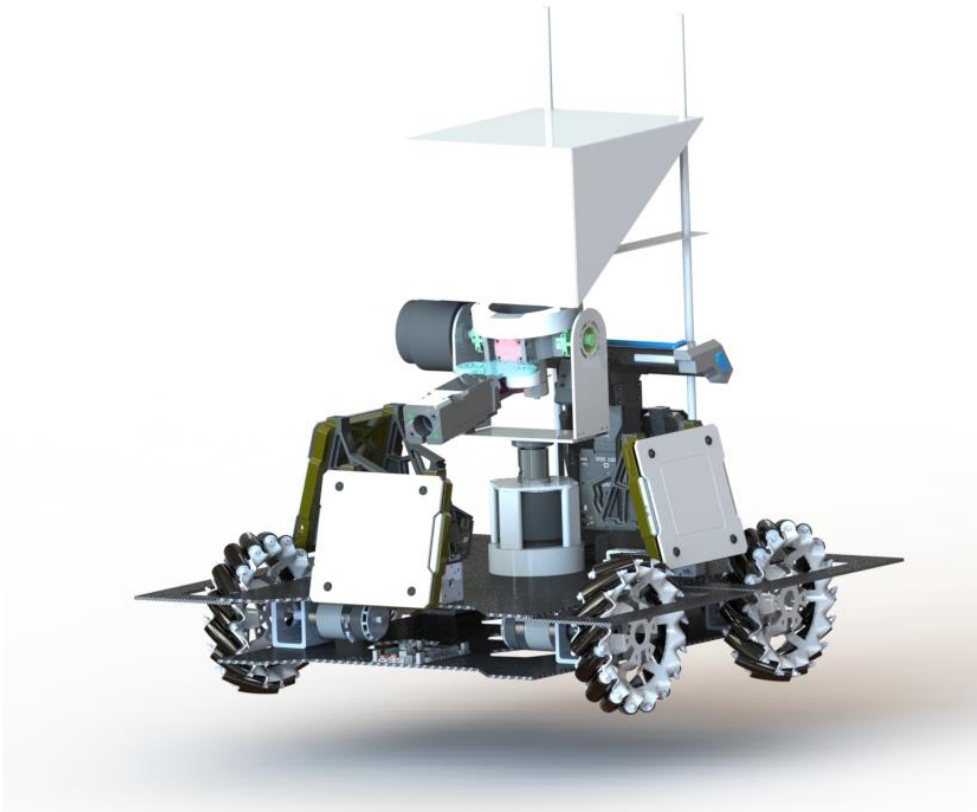
机械部分设计亮点：

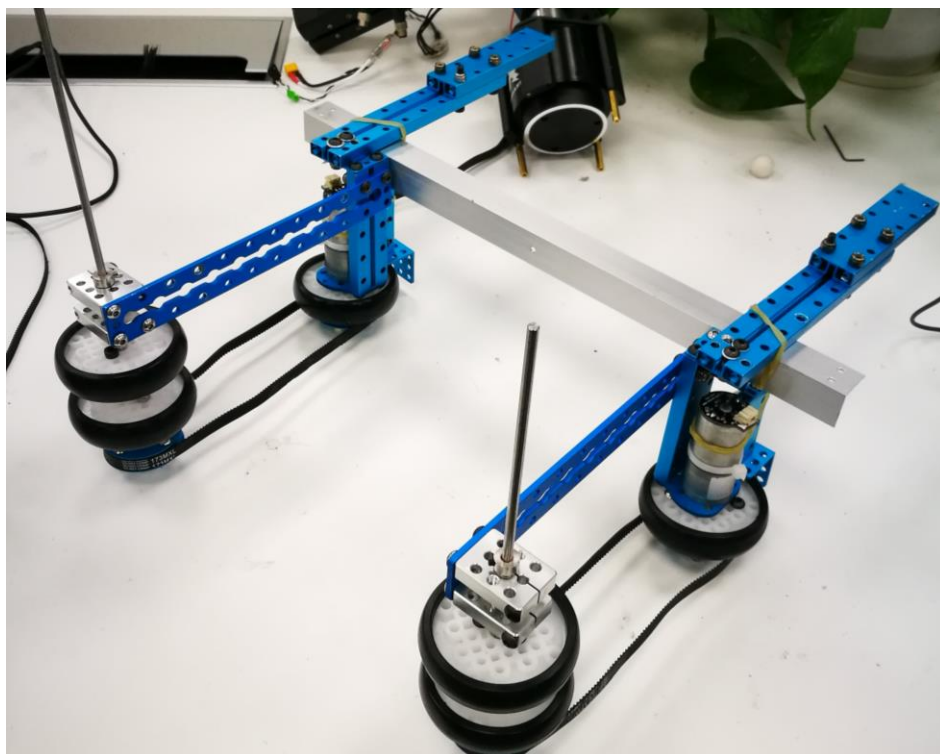
- 1 分体的弹仓设计：云台可以不用承受过大的重力。
- 2 轮收设计：高效，迅速，容错高。
- 3 全包围防护板：防撞。含导轮，减少侧面摩擦，不易损坏。

算法部分：尝试使用颜色识别加强击打能力

2、机械设计

2.1 项目模型

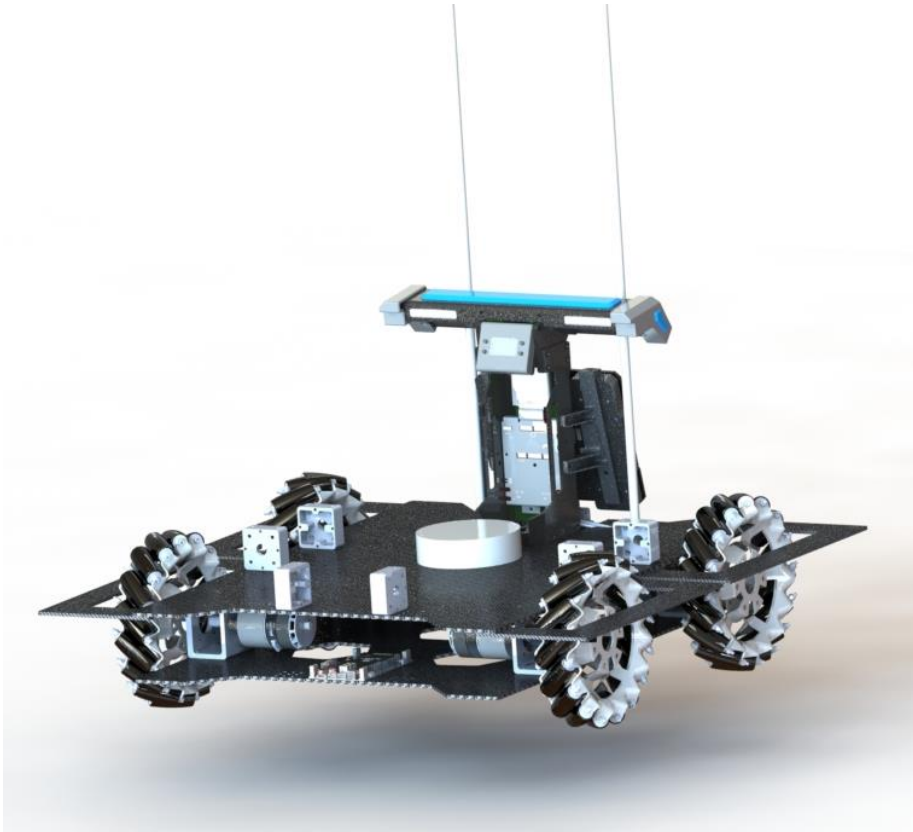




此台机器人是 RM 步兵车和机械爪的结合，它的功能其实就是步兵车的功能加上一个夹取物块挡住一个装甲板。我们机器人的最终版中没有加入机械臂,因为其实在场上它并不能发挥他应该发挥的作用。

2.2 各模块设计思路

2.2.1 底盘



我们所做的机器人是 RM 步兵车机器人和一个抬块模块的结合。

因为 D J I 给了一部分配件，其中 3 5 8 0 电机支架的孔位最适合使用双板底盘因此我们选择了这种结构。

最先考虑到的是板间的固定形式，我们以电机支架为主体的同时，在两侧拿碳板和 m a k e b l o c k 连接件固定。

为了防止发生陷入场边缘隙和因冲撞而使底盘碎裂的情况，我们加了四块可更换的防护板。

麦轮的安装刚开始使用了如图所示的内八字安装法，但经分析发现，虽然这样的安法可以极大地增强机器人的机动性，但是对于步兵机器人来说，这种安法会让他在原地旋转时大打飘，因此在之后采用了外八字安装法。

上下底板的建造是使用 s o l i d w o r k s 建模，模拟装配确认孔位无误

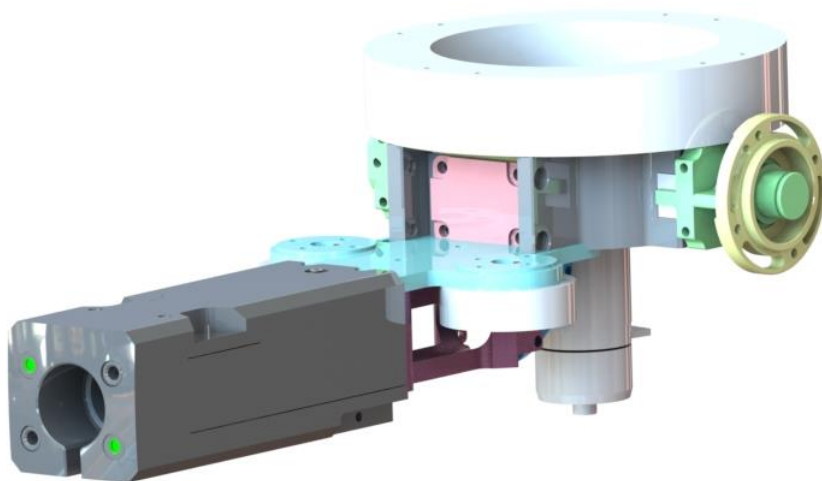
后，用雕刻机雕刻碳板而成的。

2.2.2 云台



我们完全参照了RM光放给的图纸。但由于云台上部没有给配件，所以我们又用碳板废料进行了切割，这一过程耗时很久，因为我们已经用了三块儿碳板，无法在第一时间使用雕刻机。这也在一定程度上导致我们整个进度的滞后。我们的云台在上底板上的安装位置偏后是考虑到射弹系统重心靠前，并且还想要在前方加一个块儿的结果。

2.2.3 射弹模块



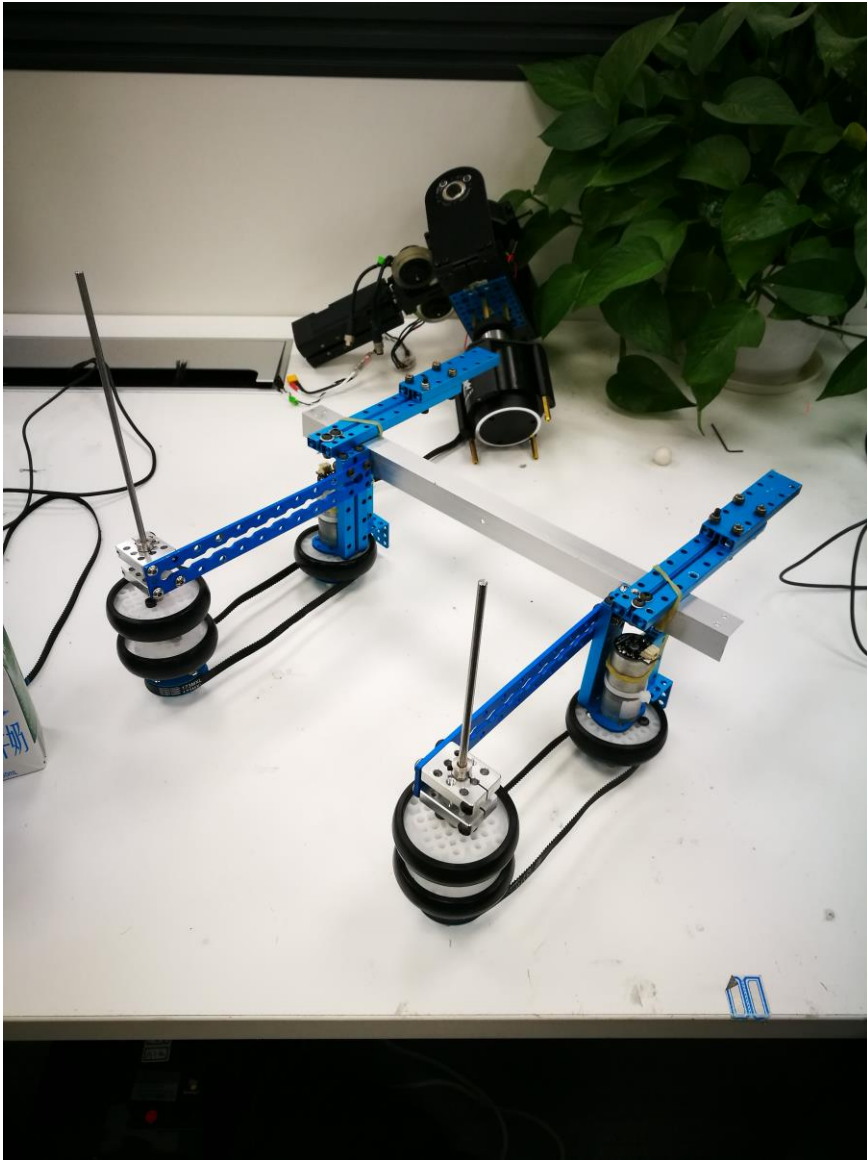
我们选择了上供弹弯管射弹，我们考虑到时间和材料都不允许我们做下供弹，再加上这次射击主要针对下部装甲板，因此我们选择了弯管射击。此模块与云台之间的连接件有我们自行建模并 3D 打印而成。

2.2.4 弹舱



考虑到雕刻机切割时间过长，我们选择用既不软又不硬的复写板作为制作弹舱的结构。

2.2.5 抬块模块



我们刚开始有两种想法，一种是做一个夹子夹取，一种是用轮子收。考虑到要挡住装甲板，我们需要抓住块以后抬起，而夹子在这方面会很不稳定，于是我们选择用摩擦轮收取，然后用舵机带动大臂抬起。

2.3 优化和改动

2.3.1 底盘

由于考虑不周到和之后所关联结构的变化，我们又对孔进行了填

补和扩增。第一版下底板的减重孔由五个六边形组成，之后因为板子打坏了，我们对板子进行了调整，我们把两块防护板加入这样可以充分利用这块板子，节省材料，单打出来以后发现强度不够，于是我们换回了之前的正六边形设计。也是因为这个，我们不能再频繁地使用雕刻机了。

2.3.2 云台

因为设计的原因，我们的云台重心太过靠前，因此我们用皮筋在后面给云台施加了一个力，是云台在不受电机力的作用的情况下可以保持水平。本着“没有什么是一根橡皮筋干不了的，如果有就加一根”的原则，我们家了 8 根橡皮筋。加橡皮筋主要是为了提升云台的机动性，因为云台电机的力其实足够大，总是能到你想要的位置的。

2.3.3 射弹模块

没有进行什么大的改动。

2.3.4 弹舱

我们最终也没有调整好这个模块，但我们进行了问题分析，机械结构这块的问题在于以拨盘底边竖直向上的这部分区域过高，导致弹丸对拨盘和弹丸的压力过大导致无法正常射弹。还有可能是我们加的弹舱与射弹模块的连接装置给了弹丸一个向下的压力。排除这些原因，我们在做试验时发现，即使装很少的弹丸，依然会卡弹，我认为这就有可能是程序的锅了。

2.4 可行性测试

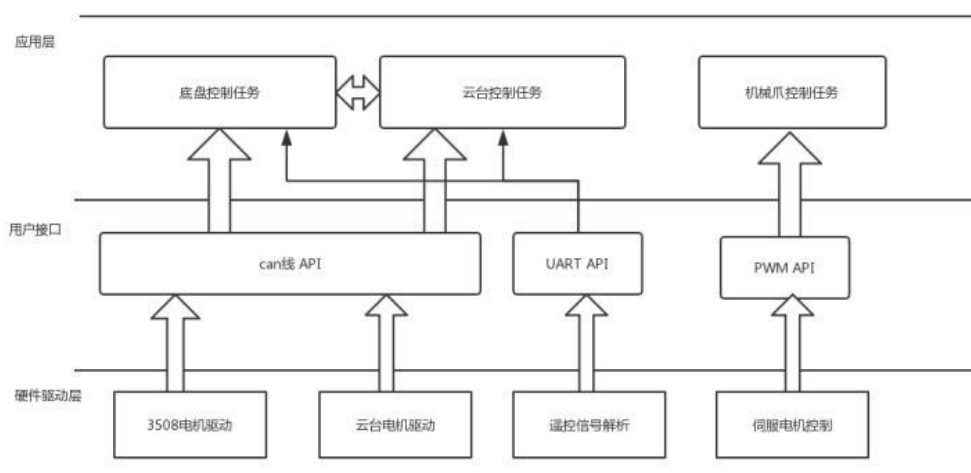
底盘：做出地盘之后，我们到场地跑了一圈，把所有的坡都跑一边，保证不翻不磕。

机械臂：我们搭出了一个用轮驱动的机械臂，与嵌入式接线通电一起测试。箱子能很容易地被收进机械臂，并且被夹得很紧，电机反转也能很容易的被推出。装上舵机之后它也能成功被抬起。

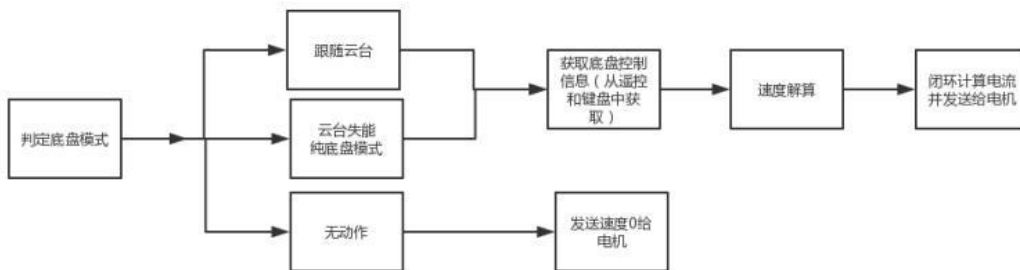
3、控制部分

3.1 项目软件架构

3.1.1 软件框图



3.1.2 操作说明



3.2 云台模块

问题一：云台 pitch 轴抖动

解决：查看手册，确定 IMU 的方向，对照程序中调用的是数据确定传感器的摆放，并将传感器尽可能摆放在 pitch 轴轴心上方。

问题二：因为我们弹仓的设计缺陷，导致云台在自然状态下无法平衡。所以当云台复位时，阻力过大。当误差比较小时，调节的力量小到不足以克服阻力，一直不能完成复位。

解决：在云台后加装皮筋平衡阻力，同时修改初始化函数，加大判断复位完成函数的阈值，使云台能早跳出初始化过程，进入受控状态。

问题三：

将视觉调节的用控制添加到云台控制中。

解决：方案一：模拟遥控信号，将控制量作为遥控信号叠加。

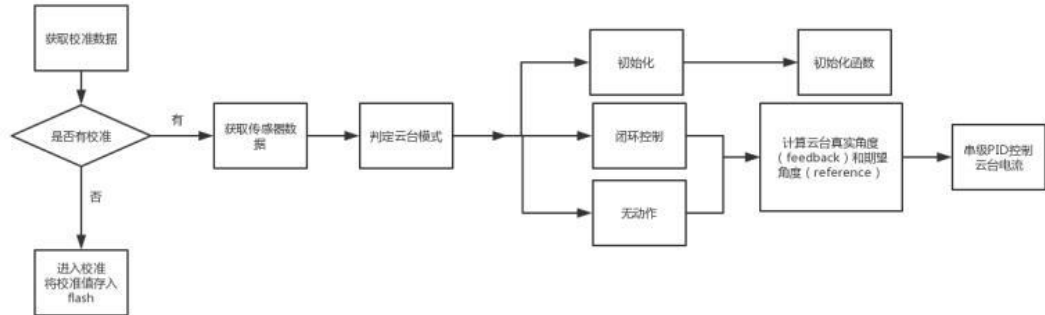
但效果并不理想，偶尔能实现跟随。

反思：将视觉控制量叠加在期望值是不正确的，因为此时云台朝

向并不一定指向期望值，所以叠加控制会有误差。

方案二：

将控制量乘以一个比例系数叠加到云台真实朝向上做 PID 控制。



3.3 底盘模块

问题一：最开始使我们的车为 o 行装配，与官方的例程不符，小车不能实现全向移动。

解决：上网查阅资料了解速度解算的推导，手算出 o 型速度解算公式。

3.4 发射模块

问题：由于弹仓设计缺陷，拨弹电机拨轮上方压力过大，经常出现卡弹问题。

解决：修改拨弹电机期望速度的 PID P 部分参数，使拨弹电机转速变小。让子弹有更多的时间落入摩擦轮中发射。

衍生问题：当弹仓压力过大时，由于拨弹电机转速过小，负载能力变

差，有可能卡住不转。（待解决问题）!!

4、算法部分

4.1 对抗赛算法设计

项目名称：装甲板检测系统

1. 功能需求：检测场景中地方机器人装甲板的位置，将旋转信息发送给云台，实现对目标的跟踪，起到辅助瞄准的作用

2. 其他要求：①在目标旋转或快速移动时保证检测的稳定性

②尽量减少系统延迟（通过优化算法）

3. 整体构架

(1) 文件描述

Camera.cpp: 独立文件，用于标定摄像头，计算相机内参

uart.cpp: Linux 串口通信，用于在上下位机间发送数据

RoboVision.cpp: 对装甲板进行检测，通过算法确定最终云台需要转动的角度，并通过串口通信发送给下位机

main.cpp : 程序入口

(2) 程序逻辑：系统启动后，摄像头开启，以 120 fps，一帧一帧对装甲板进行检测，通过算法确定最终云台需要转动的角度，并通过串口通信发送给下位机

4. 实现方案：

摄像头：购买的摄像头模组位 K2SA17（90 度无畸变），支持以

120fps MJPG 640*480 摄像。用 3D 打印件对摄像头模组进行保护，并用壳将摄像头安装至图传上方

运算平台：搭载了 Tegra TK1 的 DJI Manifold 妙算平台，运行 ubuntu 系统，固定在车体上，通过 4Pin 线与开发板连接（需将 4Pin 线的 Tx 和 Rx 对调）

第三方库支持：OpenCV > 3.0

运行效果：绿色方框位最终确定要击打的装甲板

5. 检测部分算法：

(1) 由 R 通道与 B 通道作差，以凸显蓝色或红色（蒸馏图像）

(2) 对蒸馏后的图像进行二值化处理，得到场景中特定蓝色物体对原图像的灰度图进行二值化处理，得到场景中特定亮度的物体

(3) 对两张二值化后的图像进行按位与操作，得到场景中蓝色与高亮的公共物体，即灯条蓝到发白的部分和蓝色光晕部分的轮廓，对找到轮廓进行矩形拟合，得到单条灯柱的部分

(4) 运用约束条件排除形态差异过大的灯条以及显示血量的横着的灯条，由灯条之间的关系判断两两灯条是否能够构成一块装甲板，由灯条位置得到装甲板位置，并确定装甲板中心点位置，对装甲板进行排序，选择较大的装甲板作为最终装甲板

(5) PnP 解算，计算摄像头相对于目标位置的姿态（需要知道相机内参，提前对相机进行标定）

(6) 由于云台只有 yaw 轴和 pitch 轴，故可以在一个平面上考

考虑 yaw 轴的旋转角度，计算出旋转角度，通过串口通信发送给上位机

(更详细内容请见附件)

5、总结

5.1 项目收获

嵌入式有话说：

实现通过串口与上位机通信，实现 PMW 控制舵机，实现步进电机的控制，实现对直流电机的驱动和换向……

对很多控制器如步进电机驱动器，直流电机驱动板等陌生，不会使用。解决：上淘宝查阅资料，依据手册编写控制程序。发现自己原来很惧怕陌生的知识，总给自己寻找蜷缩在安全圈中的理由。和同学们在冬令营相处的这几天中，我看到了每个同学身上值得我学习的闪光点，看到了他们对科技的痴迷，对新知识的渴望。我突然发现，那些我原来认为高大上、难以理解的知识在我认真系统的学习下变得如此简单、显然。求知永远是一件快乐的事。

算法：

这次冬令营中,我挑战自我,实现了视觉识别,对自身编程能力和自主学习能力有很大的提升,同时对机器人有了较为系统的了解,也吸收了许多团队合作方面的经验,期待有机会的话,在夏令营再

次挑战自我,做出优秀的项目

机械：对于 solidwork 的使用，我们用得更加熟练。对于机械设计，我们有了更深的了解。学会如何与队友沟通。

5.2 马后炮

1.机械需要增加人员比例，算法则需要降低，嵌入式更多时候需要两位人员轮换。一个完美的项目是机械程序电子的完美配合，一个环节都不能缺失。机械程序电子没有水平高低之分，只有大家齐心协力才能孕育出一个完美的作品。

2.队伍内需要需要更多的沟通，嵌入式和机械需要同时沟通推进，而不是独立进行，大部分队员也需要去了解其他领域的知识，也只有这样队内交流沟通时才不会存在障碍。

3. 本次冬令营的任务需要硬件、软件以及多个系统的配合，即使编写出了较好的算法，没有好的硬件，也是无用的，只有软硬件相配合，才能构建出更优秀的系统

机械：首先我们会先了解实验室的加工技术，了解周全了以后进行讨论，拉上嵌入式各出各的主意，确定方案后再进行画图，这时的分工会变为一人画图，两人做实验，对我们设计的结构进行评测，之后加工零件进行组装。

对于一些结构我们会进行取舍，来让机器人达到在场上的最优

状态。例如弹夹大容量我们刚开始想的是 700 颗，但其实 400 颗就够了，这样既不会卡弹又射的准。