

2022RoboGame 壶壶绝杀队技术总结

写在前面

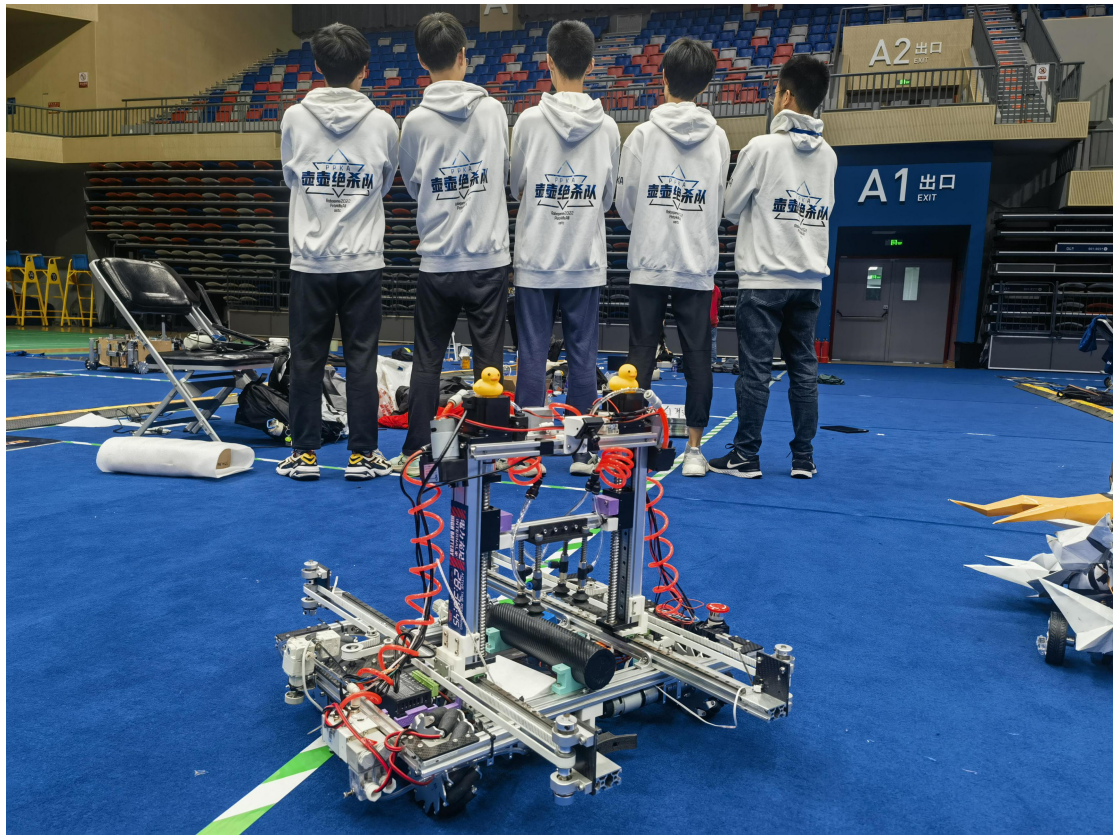
这里是壶壶绝杀队(AKA.PPKA)2022RoboGmae 技术总结。

参赛项目：冰壶机器人

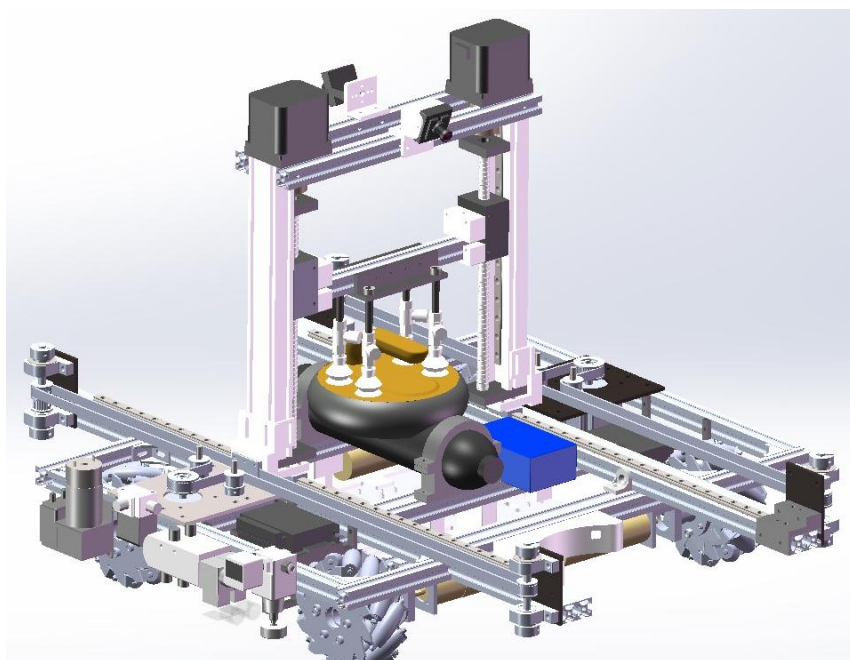
参赛队员介绍：

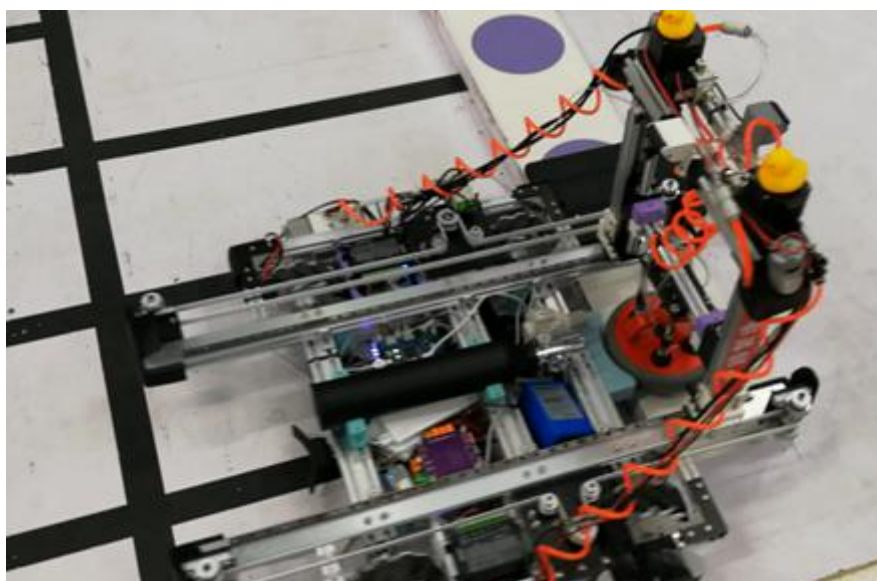
姓名	学号	学院	邮箱
林文浩	PB21050974	工程科学学院	lwh1767170750@mail.ustc.edu.cn
刘思源	PB21061235	信息科学技术学院	liusiyuan0524@mail.ustc.edu.cn
张昊鹏	PB21050991	网络空间安全学院	dinasour687@mail.ustc.edu.cn
卢星宇	PB21061239	信息科学技术学院	luxingyu010@mail.ustc.edu.cn
蒋宇航	PB21061216	信息科学技术学院	jyh210683@mail.ustc.edu.cn

Overview



机械





为了适应本次比赛任务的特殊性，我们队在机械结构上进行了很大的创新，我们将移动型龙门架和移动底盘有机地结合在一起，使得我们的机器人能够在不转身的情况下对前方和后方的冰壶都能进行拾取，大大提高了机器人抓取冰壶的灵活性。其实也没快很多但是很炫。

同时，我们队率先性地开创了采用气动装置发射冰壶的方法，这种方式在比赛中被证明是十分稳定的，而且大大降低了制作成本。除了在大体结构上进行有效创新，我们还在各项细节处不断打磨，追求让机器人达到最好的效果，比如我们特意让真空泵吸盘尽量距离近，保证气路上漏气的现象明显减少，而且为了防止单个真空泵故障导致失误，我们还设计了双真空泵的结构，这样子即使一个真空泵发生故障而不工作，机器人依然可以完成各项任务。再比如我们还在车上挂有额外的电磁阀，当一个电磁阀坏掉的时候依然可以迅速替换备用电磁阀继续工作。

车身总体的设计过程中，我们还考虑了各种力学因素，比如如何有效降低全车的重心，来保证车身运动平稳，比如各 3d 打印件的结构都加上肋板，用来增加零件的抗弯截面系数和抗扭截面系数。

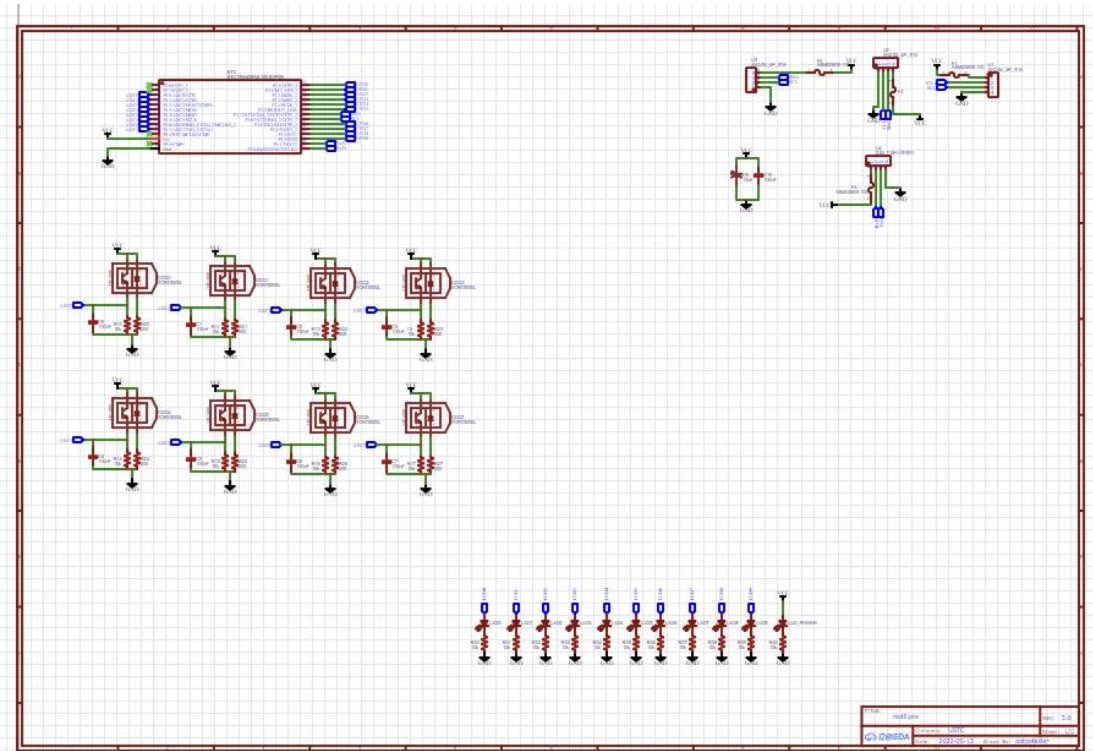
下面是一些我们搭建机器人过程中踩的坑，供后人参考以免重蹈覆辙：

1. 如果采用真空吸盘或者其他需要用到负压气路的方案，需要注意电磁阀得用特制的真空电磁阀，不能用普通电磁阀
2. 如果采用气路，需要注意在每个接口缠上生料带，买回来的接口上面本来就有白色的那几圈是没用的，要自己缠。不然就只能听到嘶嘶嘶的漏气声。
3. 如果车比较大比较重，尽量选大一点的轮子，不然到时候肯定会后悔。而且轮组电机的连接方式也要注意，尽量选电机的轮子不直连的方案，否则由于重力影响轮子歪的严重。
4. 车架横竖的铝管不要在同一平面内连接，最好分上下层，否则车的刚度不足，会变形。

电控

红外传感器

沿用了 HelloWorld 队的设计方案，并进行了一系列没有任何意义的修改。使用 c51 的 ADC 读取红外传感器，并通过 I2C 协议发送数据给主控板。



软件模拟 I2C 和 stm32 的硬件 I2C 的 hal 库实现在通讯时出现过一系列奇怪的问题，如 51 速率跟不上 stm 的时钟信号、接线不稳定毛刺导致时序错乱、hal 库死锁无法通信等等扰人的问题，但是其原版方案（使用单独的 stm32 来接受 I2C 再转发给主控）似乎没有出现问题。

可以上 stm 的芯片方便一致开发，建议使用硬件 I2C，可以尝试更快的 spi 进行通讯。

主控 stm32F407ZGT6

首先是血泪教训总结出的注意点：

1. 注意单机电源连接。尽量避免单片机直接接入降压后的电源电源供电电路。电机等大功率器件运动时将对电源纹波造成很大影响，这样的电气效应将会对一般核心板上的降压芯片造成很大压力，直至烧坏降压稳压芯片。（血的教训，我们烧掉了两块

单片机才找到原因)。我们推荐使用充电宝直接对核心板供电。

2. 注意 HAL 库的陷阱。我们在临近比赛的时候遇到了一个致命问题，单片机和香橙派的串口通信不稳定，有时候两者间的串口通信会被直接打断并不再重新启动。这是极其致命的。最后我们在 STM 的论坛找到了可能的原因——半双工外设的进程锁(可能叫这个，也可能不叫)相互抢占并最终锁死。建议同学们对 HAL 生成的文件进行一定修改。(问题不只这一个，大家要敢修改 HAL 库以实现功能)。

3. 注意用电安全，车辆有大量导电材料如铝型材、碳板等等，请注意使用不导电的材料进行保护，请区分用 24/12V 高压电和 5V 普通电，建议使用不同线缆区分，不要把 24V 电通进单片机一波带走，也不要一不小心碰到金属劈里啪啦。

接下来是本次比赛中的解析：

本次比赛中我们的主控模块采用了来自反客科技的 STM32F407ZGT6 最小系统板(想要的可以自己上淘宝直接搜索)。这家的板子做工不错，店家客服态度很好。推荐购买。

本次的程序我们开源在 Gitee 仓库，仓库链接为 <https://gitee.com/pot-pot-kill-team/ppka-core-board-program--hal/tree/master>。大家可以直接 clone 使用。

在本次的代码中我们采用了面向对象的思想，不同对象的实现之间分离并使用规范好的接口进行调用。因为如此，我们本次比赛代码可以在简单修改后直接被套用到别的项目之中。比如四轮小车，更比如之后的 RoboGame 比赛之中。欢迎大家交流，大家的问题都可以通过写在开头的邮箱与我们的相关队员取得联系(本电控部分联系张昊鹏，卢星宇，刘思源)，我们将尽力提供帮助但不再维护 Gitee 仓库。

代码与电机等设备交互的底层库编写较为规范，上层仓库引入了一些全局变量可能造成代码阅读的一些障碍(也可能是福利)。

本次的通信函数也进行了开源，协议在 gitee 仓库中可以找到。我们通过该协议实现了较为稳定与准确的通信。

本次代码的一些不足是使用了一些阻滞式的函数，没能合理使用外设的特性，导致了一些性能的浪费。这虽然没有对本次的比赛照成很大的影响，但改进将会带来很大的提升。

本次比赛的代码在比赛中稳定运行，实现了我们的编写需求。合理的架构使上层调整迅速，上层新增功能快速，为我们节约了很多的调试时间。

陀螺仪 Jy901s

无法用程序发送指令进行加计校准，我们的锅。

整体上相当稳定，推荐。

香橙派

买草莓派不如买橘子派，买橘子派不如上 Jetson，Jetson 不如 3070 笔记本。



让你感受老黄的神奇力量。

行程开关

我们用热熔胶把它黏在一些地方，但是比赛时掉了没发现，然后比赛就没了。建议好好固定车辆的任何一个部位。谁直线不会加油啊，弯道快，才是真的快；谁比赛不会搭车啊，不掉件，才真的行。

视觉

我们视觉做的相对比较多，主要在以下方面：

1. 冰壶的识别

冰壶的识别我们采用传统的颜色提取算法：

- (1) 先将图片从 RGB 颜色空间转换到 HSV 颜色空间
- (2) 利用调好的 HSV 空间下红色和黄色的参数分别提取两张阈值分割的图片
- (3) 利用形态学变换进行噪点的去除和内部小孔的连接
- (4) 计算两张图中亮的部分的面积，面积的大小即对应红色和黄色的颜色的多少
- (5) 比较两个面积的大小，通常视线里只有红壶或黄壶的时候，两个面积的值相差很大，当没有冰壶的时候则两个面积都很小，以此来判断视线中冰壶的颜色

2. 视觉辅助机器人调整和车之间的距离

由于我们的打壶是通过和场地的侧壁发生碰撞反弹之后去打壶，这对机器人推壶的力度和角度有较高要求，力度我们通过调整机器人和冰壶的距离来实现，角度通过陀螺仪和视觉矫正（视觉通过计算来确保推壶的时候推壶的爪子是正碰向冰壶的）实现。

具体的视觉算法：

(1) 先提取颜色信息，通过形态学变换之后得到一个近似于椭圆的图

(2) 计算椭圆的中心位置和视野中心的偏移量，并将这些数据串口传输给单片机，单片机获得这些数据去控制电机运动的速度

这一部分需要进行大量的数据处理和较高的数据发送频率，而单独计算一张图片的数据时间不定也比较长（在 0.03 到 0.04 之间），为此，我们在没有计算完成的时候利用已有的数据去拟合一个函数图像，从而预测一些结果，最终到达了 50Hz 左右的数据发送频率。数据发送和计算通过 python 的多线程实现。

吐槽

因为疫情比赛推迟了两次。。。

场地木板不是平的，壶壶会乱飘。

Conclusions

强烈建议在比赛前完整检查下机器人各个部分的机械结构状况，是否锁紧，是否存在脱落，并在每次上场之前检查电池电量是否充足，建议买上一块电池作为备用。之后参加的同学希望可以放平心态，这个比赛主要在于锻炼（心态）学习技能，大家不要投入过多感情。比赛的结果（就今年的而言）很大程度上取决于运气怎么样。与其优化每一步不如多积德行善（小声）。相信爱积（ ）的队伍结果不会太差。