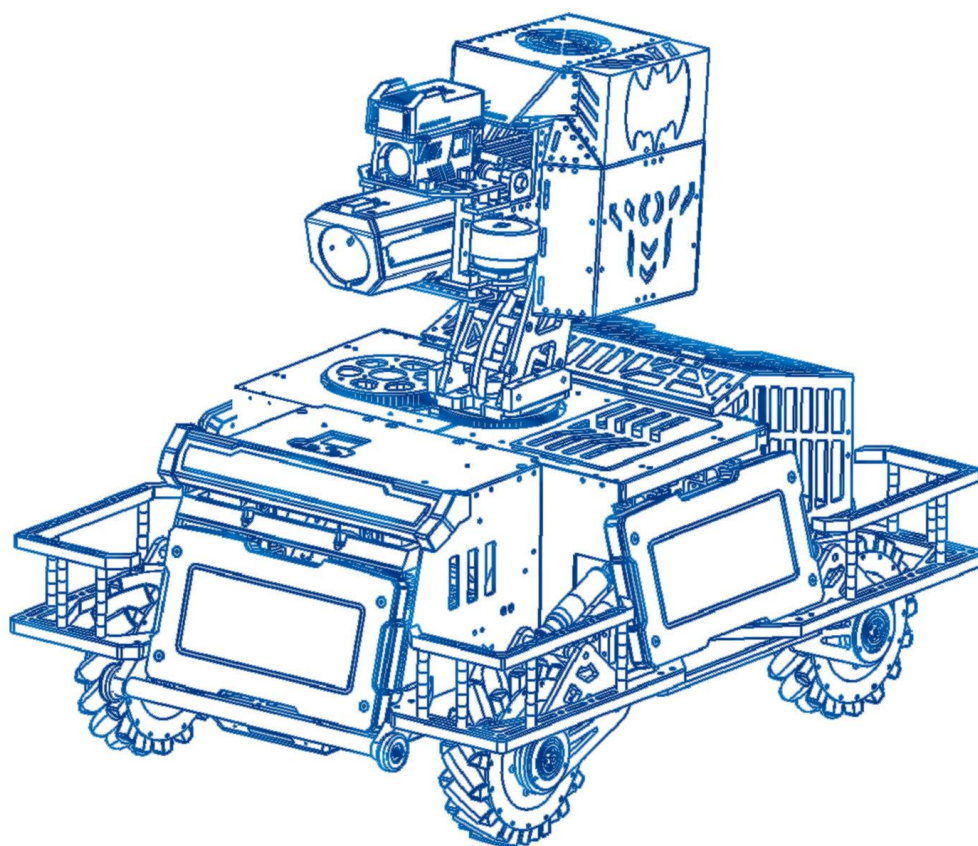


# 2023 年英雄赛季总结

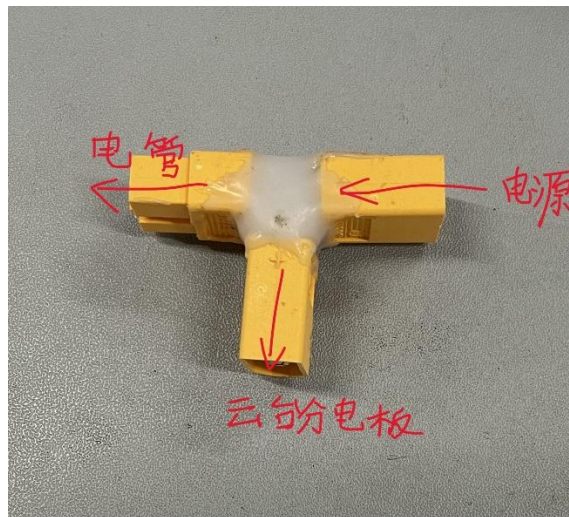


## 一、 布线

整车结构上可以被分为底盘，云台，发射机构三部分，功能上还需要承载视觉算法的小电脑（MiniPC）进行辅助。故来自电源或电源管理的电会分成四路；

电源：

由电池引出两路，一路给电源管理模块供电,另一路给云台供电（接到继电器 B 口）

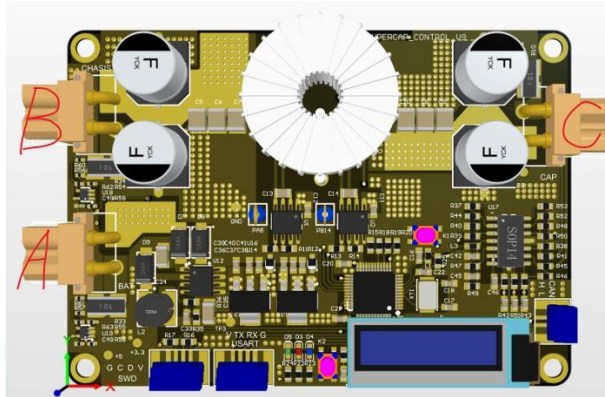


电源管理模块（电管）：

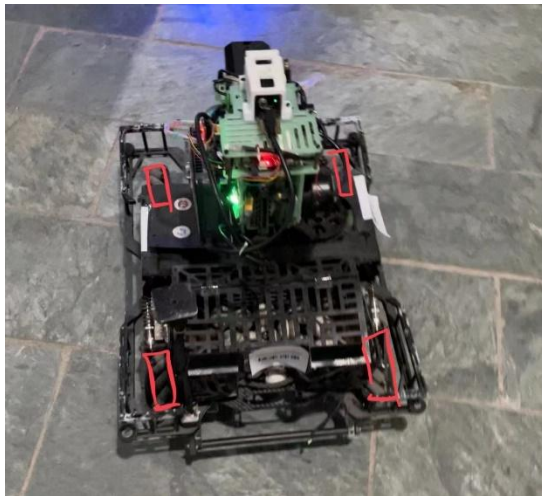


23 年超级电容 + 底盘：

超级电容控制板引出 3 个 XT 口，其中 C 口接的是电容组；



底盘是为整车提供移动能力的部分，包含 4 个 RM3508（减速比 1:19）电机。这四个电机都接到同一块分电板上，分电板再接到超级电容控制板的 B 口；



云台：

控制车上层俯仰角，偏航角，包含 2 个 GM6020 电机。Yaw 轴电机安装在机体下层，故接在下层的云台分电板上，此分电板通过滑环线给上层 Pitch 云台供电。值得注意的是，对于英雄，官方不检测其云台的功率，仅要求使用继电器通过底盘供电控制云台供电的方式来控制云台。

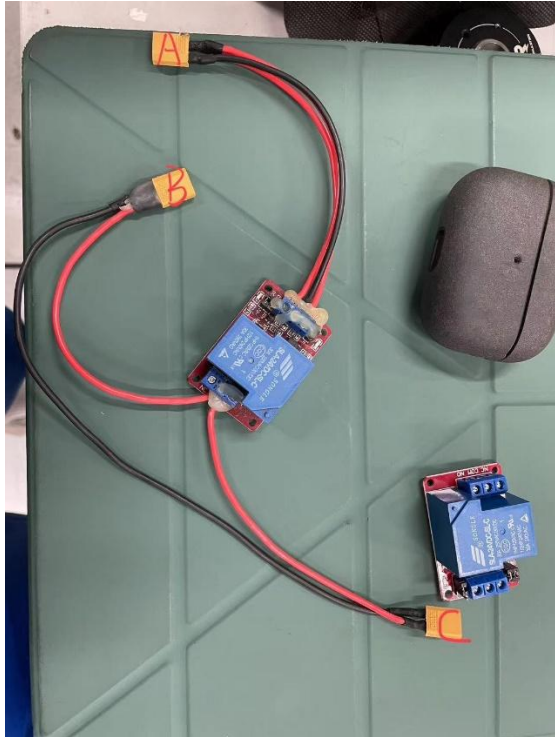


继电器的接线：

元器件原理：<http://t.csdn.cn/L1DLt>

既然要用底盘电控制云台电，那么最直观的做法就是从底盘那块分电板引线接到继电器的 A 口，但这样做其实是有问题的。假设这样做，机器人被杀死后电管 Chassis 口断电时，超级电容仍然会给底盘分电板提供一个较低的电压，这个电压虽然远不够给底盘供电，但却足以打开继电器的开关，使得云台无法立即断电，这样是过不了检录的；所以，解决方法之一就是将从电管 Chassis 口供电分两路：一路直接到继电器 A 口，一路给到超电控制板的 A 口

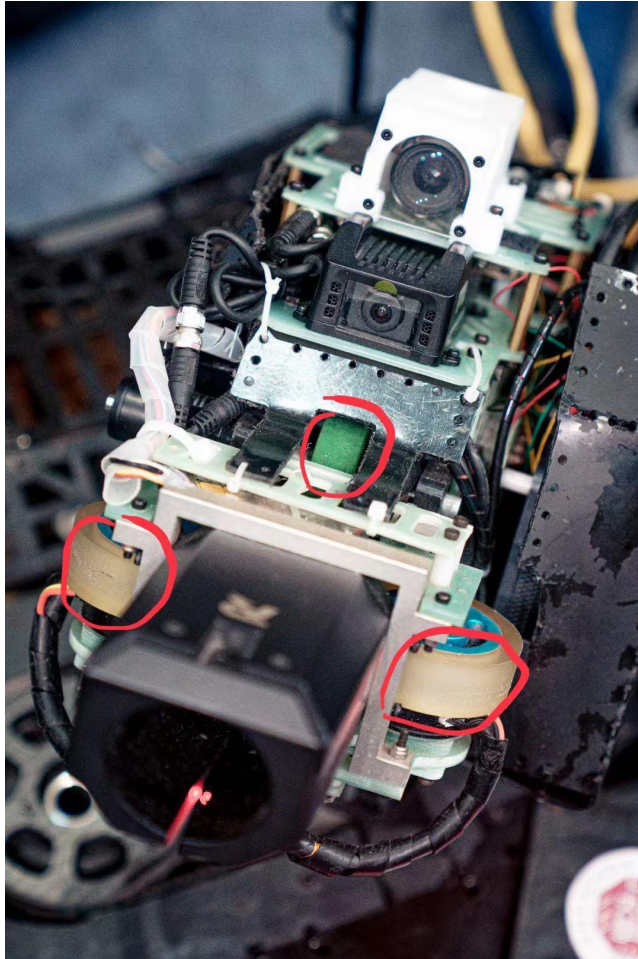




发射机构：

为两个部分，拨盘电机 RM3508 和限位电机 RM2006 为供弹部分，两个摩擦轮电机 RM3508 为打弹部分。其中，拨盘电机单独在车的下层，它接到

下层的发射分电板上，分电板由电管 Ammo\_Booster 口供电。剩下三个电机都在上层，故线都引到头部的一块分电板上，这块分电板由滑环线（接下层的发射分电板）引上来的发射电源供电。



（带动绿色橡胶轮转动的是限位电机，下方红圈两个摩擦轮由摩擦轮电机控制，拨盘电机没有标出）

小电脑（MiniPC）与主控：

由电管 MiniPC 口出来的电比较特殊，只要电源开着就不会断电（包括机器人被击杀时），故车的上下主控都靠这路供电，以获得在买活后跳过陀螺仪初始化时间的优势，当然，这意味这程序上要做一定处理来重新整车初始化。同样是这一路，其经过一个下层降压模块后过滑环线给小电脑供

电。

其它：

上主控与小电脑之间会有一个串口通讯，可以把 4 条线中的 5V 线剪掉，以保证上主控的供电稳定；

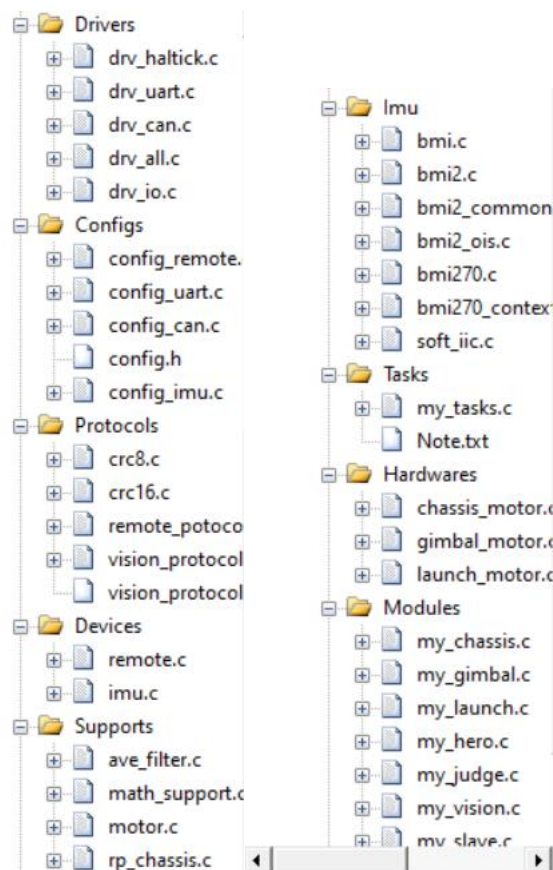
相机与电脑之间会有一条相机线用来供电与传输数据，根据经验，这条线尽量不要过度弯折（弯一点可以，绝对不要折），与电脑的 usb 接口、与相机的 MicroB 口要打胶上锁，线要买质量好的，以防止比赛掉自瞄；

有地方出问题要检修的时候，要分清优先级，比如优先检查走线外露的线材，再检查有遮挡的部分，以节省比赛中的维修时间；

线是影响车辆稳定性最大的一环之一，平时焊线不能随意，每次比赛前一定一定要检查，以防止留下难忘的非正面回忆。

代码：

上主控：



以上是从 CubeMx 生成工程后，手动拉进来的文件。其中，Drivers(驱动)、Protocols(协议)、Devices(设备)、Supports(外部支持)、Imu(陀螺仪)，基本上都是祖传，可以按需修改（陀螺仪，遥控器等）。Configs(配置) 中的函数会在接收中断中调用，用于数据更新；



剩下的内容是自己写的部分，我在国赛前重新写了一份代码，里面除了主从通讯部分每一个函数都有非常详细的注释，下面对每个部分只作重点说明。

代码：[robotpilotsEC/2023\\_GJW: Welcome to shit mountain! \(github.com\)](https://github.com/robotpilotsEC/2023_GJW_Welcome_to_shit_mountain!)

### Tasks(任务):

深大19电控陈泽帆 23/5/13

裁判系统数据、软件复位优先级最高，失联检测、视觉交互、车间通信优先级第二，车辆控制优先级最后@所有人

深大19电控陈泽帆 23/5/13



深大19电控陈泽帆 23/5/13

所有while循环必须加入超时退出操作@所有人

深大19电控陈泽帆 23/5/13

看门狗放最低优先级任务里

任务分好优先级。对于 CAN 发送，我习惯于把他们放在同一个任务里。

假如没分优先级但是又没出问题，改完优先级之后可能会疯车 qwq

### Hardwares(电机)

调用了队内的电机包，我根据不同模块创建了相应电机结构体

Modules(模块)

该部分为主要控制代码;

my\_chassis (底盘):

英雄底盘因为长宽不等长,其麦轮解算的旋转量与步兵不同,我在底盘包中作了修改;在比赛中,会要求英雄实现冲家,飞坡等对底盘性能高要求的动作,故需在代码上做处理,如直线校正,超电分配等;

直线校正: 见代码

超电分配: 23 赛季有两种使用超电的思路;

其一, 是否开启超电由操作手自行决定, 只有按下加速键时才使用超电;

其二, 超电常开; 这种情况下, 要考虑非加速行驶时的底盘限制最高速度;

比如, 可以测量在不开超电或使用较少超电时, 机器人移动的最大速度是否满足需求, 以此为标准设定最高速度。

英雄底盘非常重要, 不要止步于让它简单的动起来

my\_gimbal (云台):

云台要求的是响应快, 过渡平滑 (整定 pid 参数);

若机械上头部无法配平, 则可能需要更多的控制方法来实现;

陀螺仪的位置尽量安装在 pitch, yaw 的旋转中心;

云台是整车初始化中非常重要的一环。在比赛中, 英雄复位时云台动作, 底盘保持静止, 以防止出现大交英雄在台阶旁复位翻车的情况。除此之外, 初始化 (包括整车初始化) 还要实现超时退出, 保证云台在无法正常复位 (被卡或甚至失联) 的情况下保持底盘的可控性, 使得机器人在赛场上出状况后仍能自主移动 (回家)

my\_launch (发射):

英雄发射指标: 射速稳定性, 射频;

射速稳定性: 虽然这部分通常很大程度上取决于机械, 电控能做到的比较有限, 但还是有不少优化的方向的, 以下是个人根据观察总结的经验, 由于测试时间、次数不够多, 仅供参考;

23 年英雄发射机构有四个电机:

拨盘: 拨盘作为供弹机构直观上可能会被认为与射速的稳定性没有关系, 但实际上应该还是有的。其控制的变量在于弹丸卡进限位的位置, 最理想的情况是每次供弹, 弹丸卡在限位的位置都是一样的, 这样可以保证每颗弹丸的限位加速路程都是一样的, 进而利于弹丸出限位的速度保持一致。

限位:

采用速度加计时控制, 而非位置控制, 同样是因为要使出限位速度保持一致

摩擦轮:

摩擦轮最基础的就是 pid 的参数整定;

摩擦轮包胶对射速影响是最大的, 但其状态难以监测, 电控一般只能根据电机、裁判系统的反馈来优化。首先是电机电流, 其可以判断摩擦轮射出弹丸数, 判断摩擦轮是否堵转, 这两者可能对射速帮助不大, 但有利于优化控制逻辑。然后是电机温度, 这个反馈一定程度上可以反应摩擦轮包胶状态, 我同样上传了电机温度与摩擦轮包胶温度联系的测试数据, 可以参考。但是, 影响摩擦轮包胶的因素非常多, 我的测试数量是远远不够的, 控制的变量也可能是不全的。

射频：

23 国赛为追求高射频，做了很冒险的处理，即没有卡弹处理。24 赛季绝对不能没有卡弹处理。卡弹处理的代码是写了。因为每次供弹停转是要有堵转判断的，而卡弹堵转也会进入堵转判断，所以我根据摩擦轮打出弹丸数目来判断是否卡弹。

射频决定于你的控制逻辑。

my\_hero：

整车初始化及整车指令处理

my\_judge：

顾名思义

my\_vision：

统一发送接收频率，统一接收数据变量名以方便沟通。跟视觉，机械一起调车一定一定一定要有耐心，每个人进度都不一样，即使他不会开电池也要耐心教。联调时有问题先找自身的，不是自身的再以理服人。

my\_slave：

主从通讯部分，由于 CAN1 负载比较多，主从通讯数据量增长，故采用了降频发送。

更多细节看代码。

总结：

想把英雄调好不难，但是要究其原理或持续优化很难，希望 2024 的英雄负责人可以究原理，持续优化。