

第十八届全国大学生

智能汽车竞赛

# 技术报告



学校：天津大学

队伍名称：TJU-斩风

参赛队员：马阔 高明奇 陈邦宇

指导教师：徐天一 李祺

# 关于技术报告和学术论文使用授权的说明

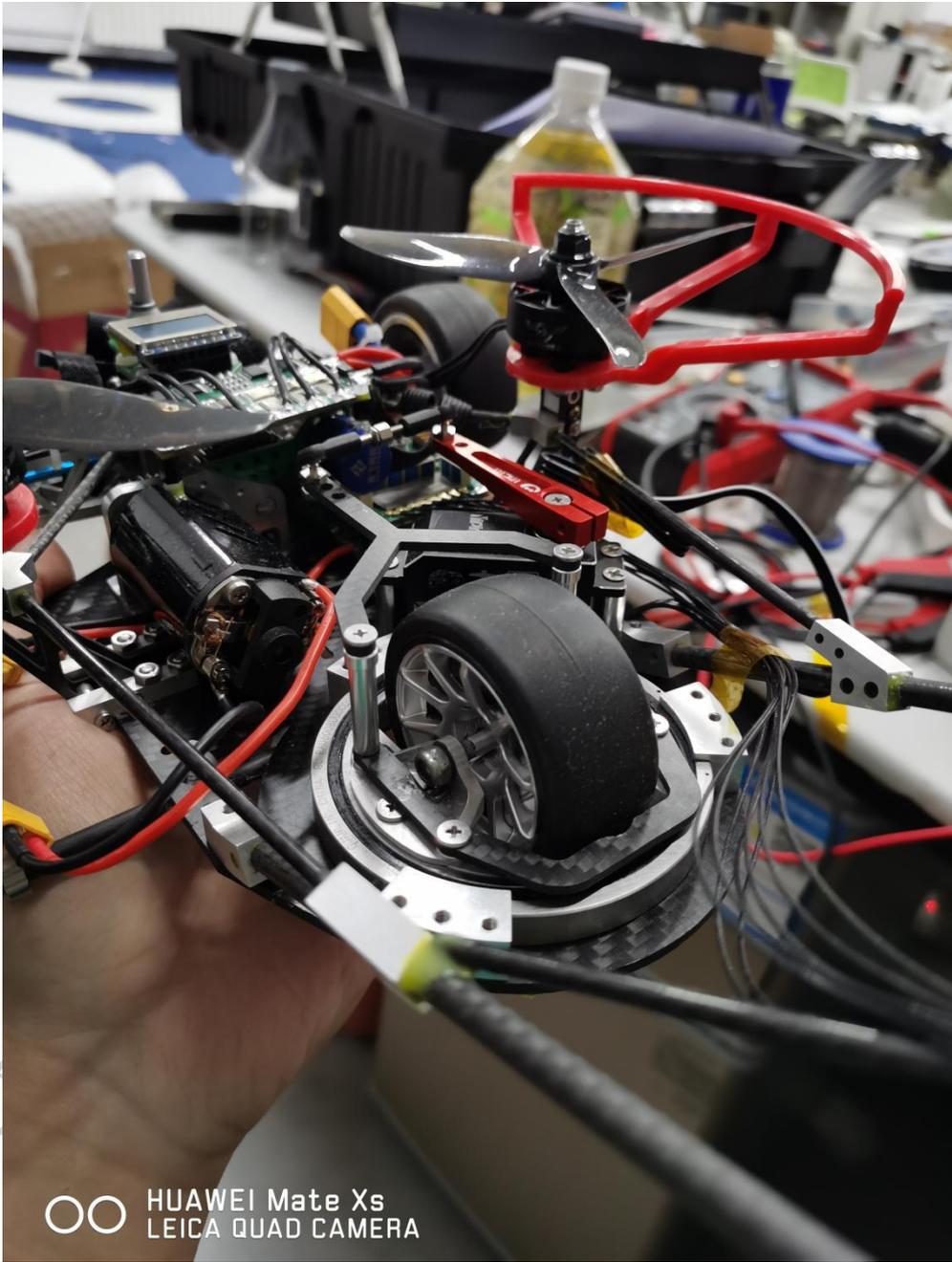
本人完全了解全国大学生智能汽车竞赛关保留、使用技术报告和学术论文的规定，即：参赛作品著作权归参赛者本人，比赛组委会可以在相关主页上收录并公开参赛作品的设计方案、技术报告以及参赛模型车的视频、图像资料，并将相关内容编纂收录在组委会出版论文集中。

参赛队员签名：\_\_\_\_\_

带队教师签名：\_\_\_\_\_

日期：\_\_\_\_\_

天津大学智能车队



**摘要：**本报告详细介绍了参赛队伍 TJU-斩风在第十八届全国大学生智能汽车竞赛负压电磁组别中的智能车系统设计。

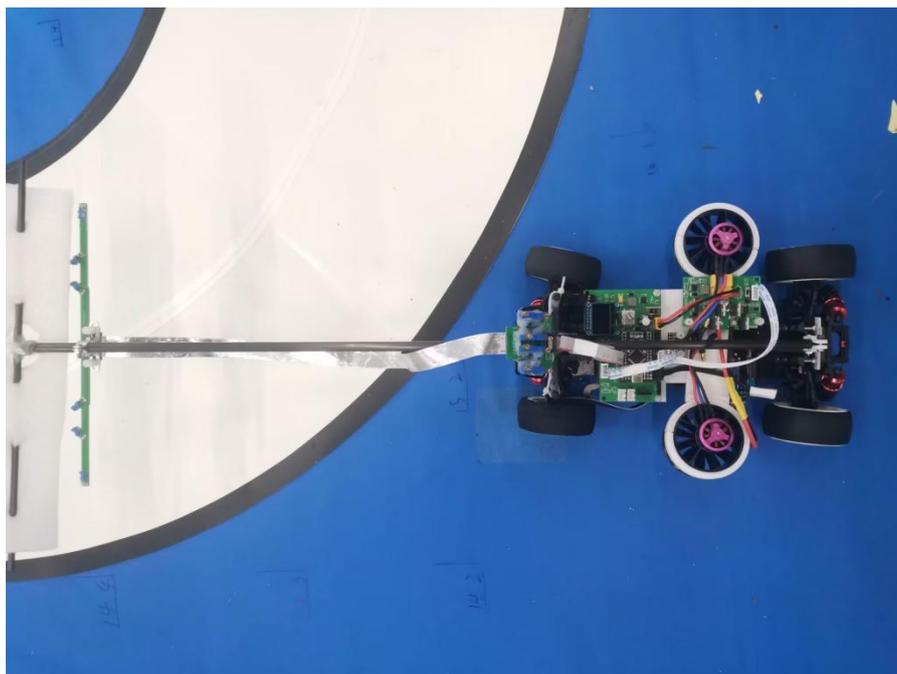
本次比赛中采用的主控芯片为 STC32G12K128 以及 STC15W408AS，整车采用符合比赛规则的全自制设计，车模采用三轮结构，舵机控制前轮转向，后轮采用单电机驱动行星伞齿机械差速器。本次比赛任务是基于 17 届基础电磁组别，采用电感循迹电磁信号线的方式来为车模提供前进引导，但不同的是可以使用负压风扇提供额外下压力，以此解决车辆运行过程中由于轮胎最大静摩擦力饱和导致的侧滑问题，以及路障、断路元素的加入。

**关键词：**智能车；负压风扇；STC；自制车模；机械差速

## 引言

相较于以往传统电磁组别，第十八届全国大学生智能汽车竞赛负压电磁组增加了负压风扇的使用限制，额外下压力的引入使得车模能够以更高的速度和更加稳定的轨迹运行，但是无刷电机等额外高功率结构的引入对整车的稳定性提出了更高的要求。早在 17 届国赛开放自制车模限制之后我就考虑过使用负压风扇方案，加入风扇之后车模运行的稳定性有了显著提高，但是也引入了额外的阻力。由于车模结构问题，导致车辆的加减速性能有所降低，最终权衡利弊在决赛前一晚拆除了负压风扇，但好在主力队友出色的软件能力，最终成功拿下线上赛国一。而 18 届负压电磁组依旧开放自制车模限制，这给了我继续完善负压方案的机会。接下来我将对车辆整体机械设计思路与加工策略、PCB 硬件电路设计、硬件结构设计、无刷电机算法设计、电磁循迹及元素处理等部分进行介绍。

不想做目录了，就这么看吧。



图：17 届负压方案

## 1 机械结构设计与实现

机械结构是智能车完成任务目标的关键所在，稳定的机械结构是软件调校的基础，也决定了车模的上限。

在 2022 年 17 届基础电磁组中我使用的就是改装过的 RC 自制车模，前后轮四驱双差速舵机转向方案。从那时起就意识到高速舵机对于转向带宽的提升，因此在 18 届自制车模的设计中考虑到保留舵机结构，但是在 17 届的调试过程中发现，由于转弯半径小，四轮四驱结构的前后差速器所需的转速差已经不可忽略。当车模采用比较激进的内切转弯路径时，往往会因为前后转速不匹配，导致中传动皮带齿上承受的负荷过高，以及前轮因为线速度不匹配原因导致的频繁跳动，使得车模稳定性降低。

在最初的设计中，我想继续沿用四驱车模结构，并增加对前后差速的分别控制。对此的解决方案可以使用机械中差按照传动力矩进行前后转速的动态分配；或者是使用双电机驱动，软件分配转速，但是双电机纵向对置的布局会给空间分配和舵机安装造成很大影响：前轮有差速器，因此无法像普通 C 车一样把舵机竖置在两轮中间；而采用卧式舵机结构，则会额外占据空间。考虑到两种方案均容易大大增加整车质量以及重心高度，并且连杆数量过多会导致转向机构回差增加，而且考虑常用 380、480 电机最佳效率点所在转速附近，如若采用一次减速的方

式，传动机构所需主齿的直径较大，会与常规前转向机构发生冲突，设计难度和结构复杂度都会相应增加，其机械故障率必然增加。在此考虑使用了单前轮转向方案，保留舵机而省略了中传动结构，拉杆数量减少到 1 根，其转向回差大大降低。

整车由于结构较为特殊，使用了大量非标结构件。三维建模仿真、数控编程、上机加工均为自行完成。接下来我将详细介绍各个结构件的设计及其加工策略。

### 1.1 前轮转向结构设计

前轮采用无动力方案，舵机控制转向。转向的实现需要使得车轮绕轴进行转动，而为了降低重心，没有使用传统三轮车结构和单车结构的前轮转轴设计。而是在设计时提出两种方案，一是采用 U 槽轴承配合铝合金弧形滑轨方案，二是使用现有标准件关节轴承方案。方案一结构相对较轻，但是零件数量多，而且由于该种非标导轨结构涉及多轴联动加工，本人所使用的桌面级机床其五联动精度最高为 $\pm 0.03\text{mm}$ ，无法满足该种程度精密配合要求，以及考虑到铝合金导轨抗冲击性能差，随着使用时间的延长容易产生变形问题，因此选用方案二，采用现有薄壁交叉辊子关节轴承，其间隙小，可以承载很高的轴向和径向载荷，并且紧贴底盘表面，重心影响相对较低。

前轮采用 1/12 55mm 橡胶轮胎，考虑旋转及载重产生的形变，因此选用 60mm 内径轴承 RA6008 承载前轮的旋转及载荷。轮胎轮轴的固定采用铝合金结构件固定在轴承内圈，通过 U 型碳板向外撑紧，如下图所示：

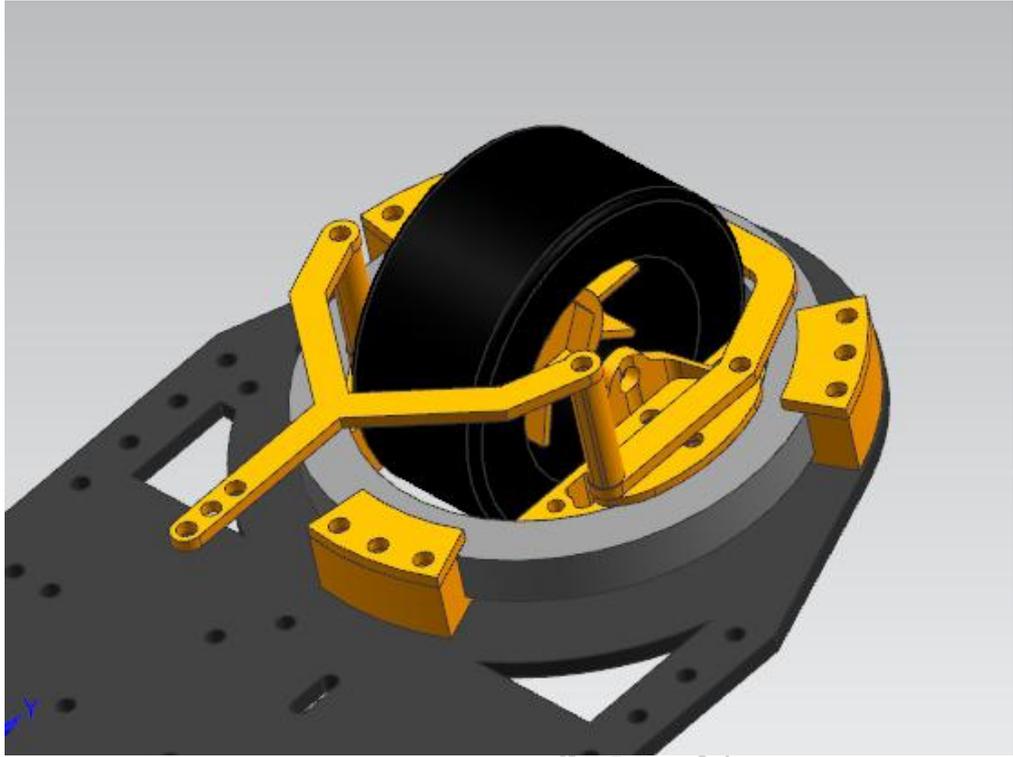


图 1-1：前转向总成

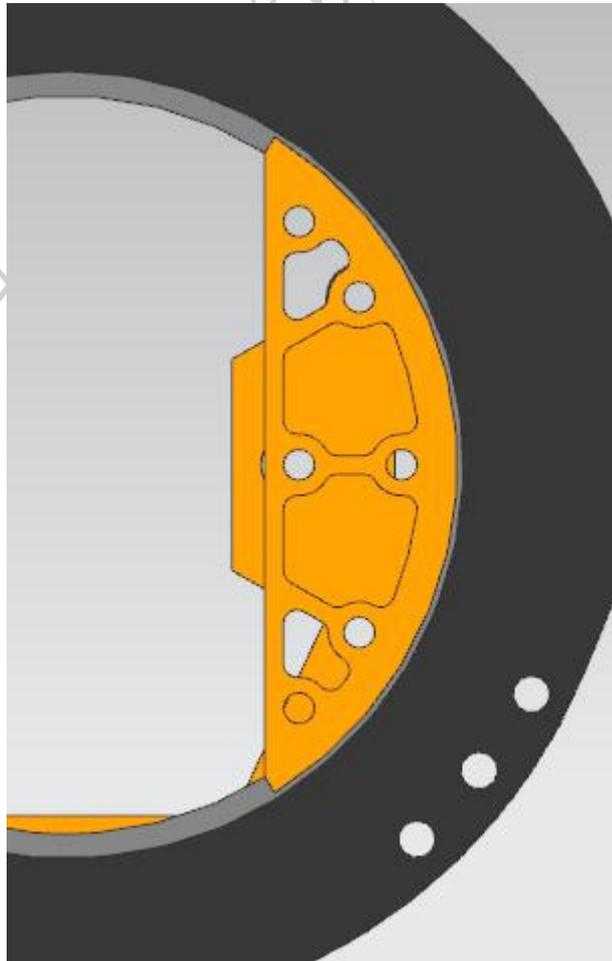


图 1-2：半杯结合器(仰视图)

半杯结合器用 7075 铝合金 CNC 加工而成，设计中采用镂空处理，其质量较轻，刚性相对较好，可以满足前轮承载的需求。在半杯结合器上钻孔攻丝，使用铝柱固定 Y 型舵机拉杆，使得舵机可以拉动前轮半杯旋转，进而实现前轮绕关节轴承轴心转动。注意这里舵机摇臂和前转向摇臂不等长，因此舵机转角和前轮转角为非线性关系，这里需在软件中作补偿。

由于前轮安装方式与传统车模轮胎作为后轮的安装方式不同，没有内侧轴承作为靠山，如果使用传统六角轮轴接合器会出现松动现象，因此设计前轮六角接合器和轮轴为一体式结构，使用 7075 铝合金 CNC 整体加工而成，该种形状零件含有轴面和多平面，适合采用车铣复合进行加工，由于缺少相关设备，在此使用三轴定轴加工上半轴体和外六角，下半轴体使用四联动进行加工，无需二次装夹，减少了拆卸装夹导致的误差。

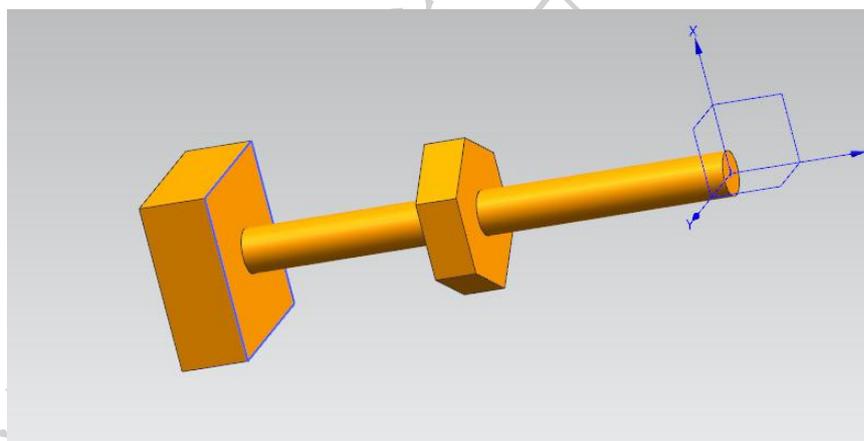


图 1-3：轮轴结构

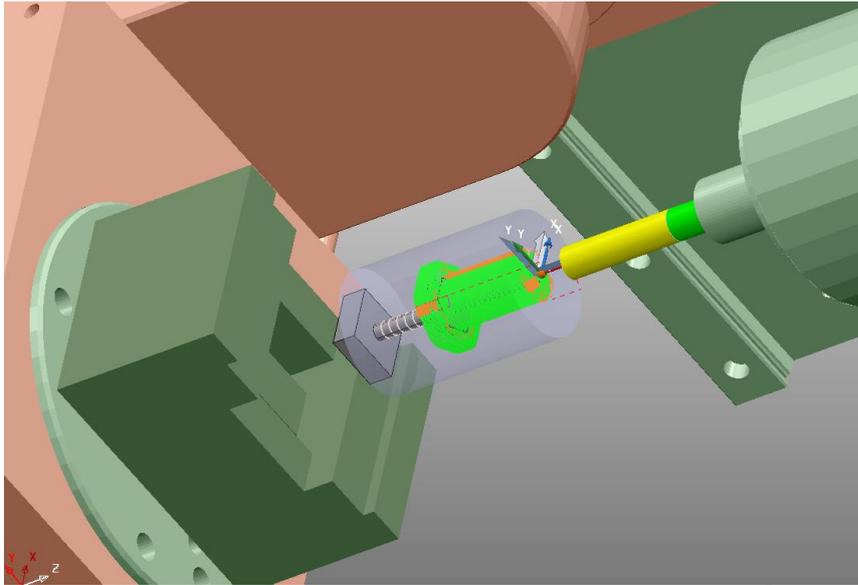


图 1-4：定轴精光

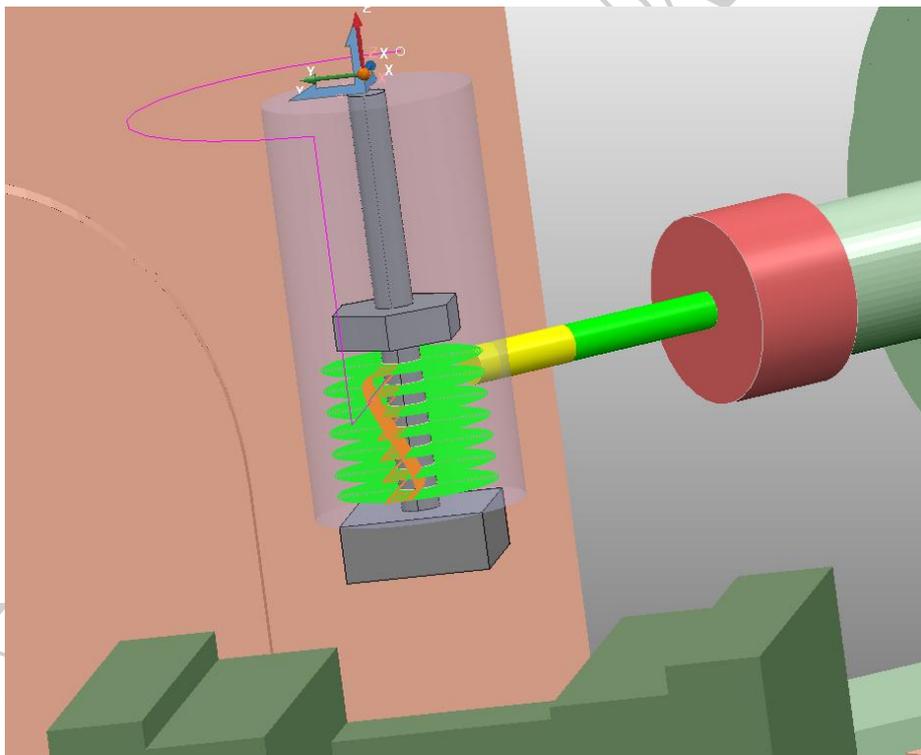


图 1-5：联动开粗

一序使用 D8 铝用钨钢铣刀，动态铣开粗到六角下端面，背吃刀量 3.0，侧 0.1，F1500，S15000，余量 0.1；精加背吃刀量 0.1，S15000，F1000；

二序使用 D4 铝用钨钢铣刀，使用镶嵌参考线加工策略四联动开粗，下切 0.1，S18000，F1200，注意立铣刀端面中心没有切削能力，绕轴旋转切削后会在毛坯

的中心区域的参考线处留下少量材料残留，因此加工完成后拆卸工件，锯掉底部方块夹持，将上半轴装夹于主轴 ER11-4mm 筒夹内，在台钳上装夹一数控车刀，主轴转速 S4000，手摇手轮车平剩余材料残留。

最终表面光洁度和公差符合要求；两侧板牙攻丝，螺母安装于轴瓦连接件内，表面浸润耐磨润滑油，转动顺畅。

前转向总成其余零件均为三轴工件，使用标准工序加工即可。



图 1-6：成品



图 1-7：前转向总成

## 1.2 负压风扇安装结构设计

18 届负压风扇的引入对车模结构稳定性和空间分配有了新的要求。在这里设计了前瞻、无刷电机支撑架、后波箱一体结构。使得前瞻抖动、无刷电机压力、后悬挂震动产生应力的均可通过碳杆沿车身多个三角结构向内传递到底盘，大大提升了结构刚性；

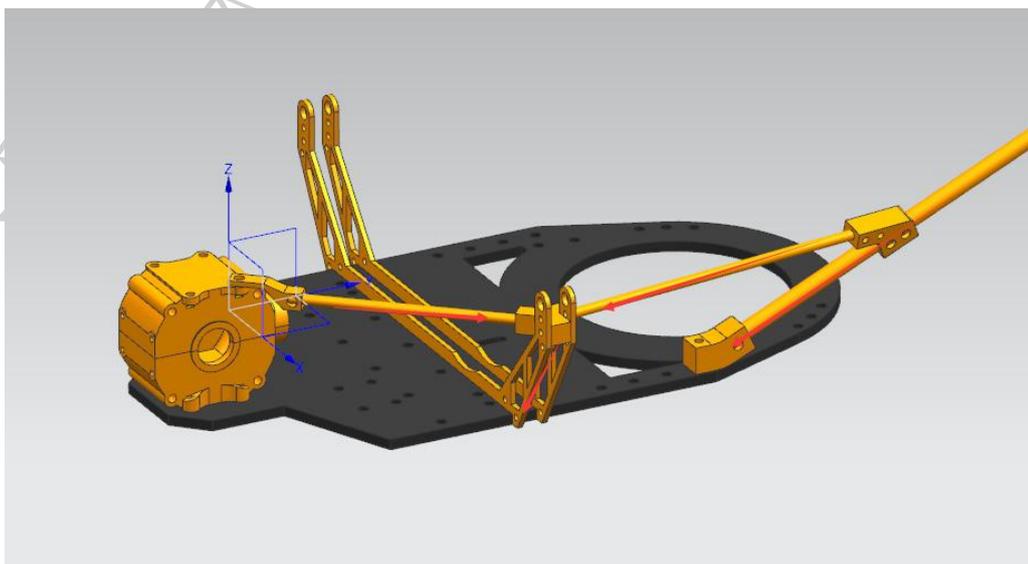


图 1-8：支撑臂结构建模

碳杆和铝合金连接件间通过 AB 胶固定,测试中通过两侧无刷电机座施加 5Kg 下压力后,整车无明显形变,后续调试过程中经过大量碰撞,机械结构(除前瞻 PCB)均未发生损坏。

但是在后续的测试中,意识到一味的追求车身刚性并不是好的选择。在很多成熟的 RC 设计中,会保留一部分车身弹性,同样使用碳板底盘但是会预留出使其可以发生扭动等形变的镂空,并且增加二楼板,降低底盘碳板厚度。因此车模在应对不那么平坦的路面时候具有更高的适应性,并且对于本车模使用的刚性悬挂,其高车身刚性更会导致车模在运行过程中轮胎不能很好的贴合地面赛道。因此这样的设计并不是最优方案,在负压电机的开启下可以缓解没有弹性悬挂导致的抓地力丢失。

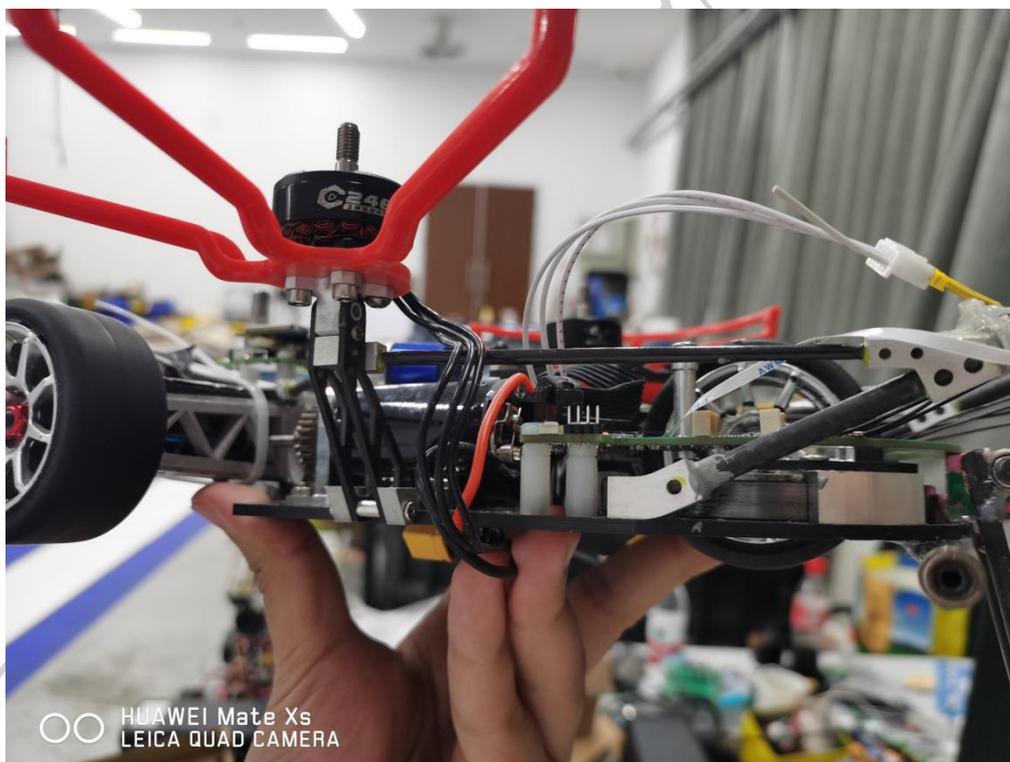


图 1-9: 支撑臂结构



图 1-10：与后波箱连接

在最初的设计中，电机固定座的设计为矢量负压方案，风扇底座可以绕轴旋转，在转向时可以通过舵机拉动连杆调节负压风扇角度，提供侧向的推力，来抵消一部分离心运动的趋势，使得转向时轮胎摩擦力更不容易达到饱和，在设计中也留了旋转轴，但是由于后期时间上的不足，于是在固定座上打另一 M2 孔，用一螺丝将其角度固定为垂直向上，移除矢量功能。



图 1-11：负压电机底座



图 1-12：矢量负压构型(未完成)

### 1.3 后驱动设计

在这里使用了 480 电机作为驱动，多级齿轮减速，行星伞齿差速，编码器使用自制连接板啮合于主齿上。后差速器使用狗骨万向节进行连接，在调节机械的过程中更换了小径铝合金狗骨，并在接杯处增加了结合器销子抗震胶圈，降低后轮传动部分由于公差及不同心度导致的震动影响。使用 100cst 和 4000cst 两种

粘度的差速油进行混合配比，调节差速器的差速阻力，使其达到相对较好的差速效果。

在高速过弯，连续加减速的情况下，机械差速相对标准 C 车软件差速的稳定性要好，电机负载相对较小，软件上可以省略一部分差速控制，只需进行速度决策即可。但是单纯的机械差速在轮胎打滑、车辆微微侧翻的情况下，转速会立刻分配到阻力较低的那一侧轮子，而另一侧轮子的扭矩会立刻下降，导致车模运行不稳定，后续如若继续使用机械差速方案的话，应考虑加入限滑功能，在两轮速度差过大时锁止差速器，以防止输出扭矩的丢失。

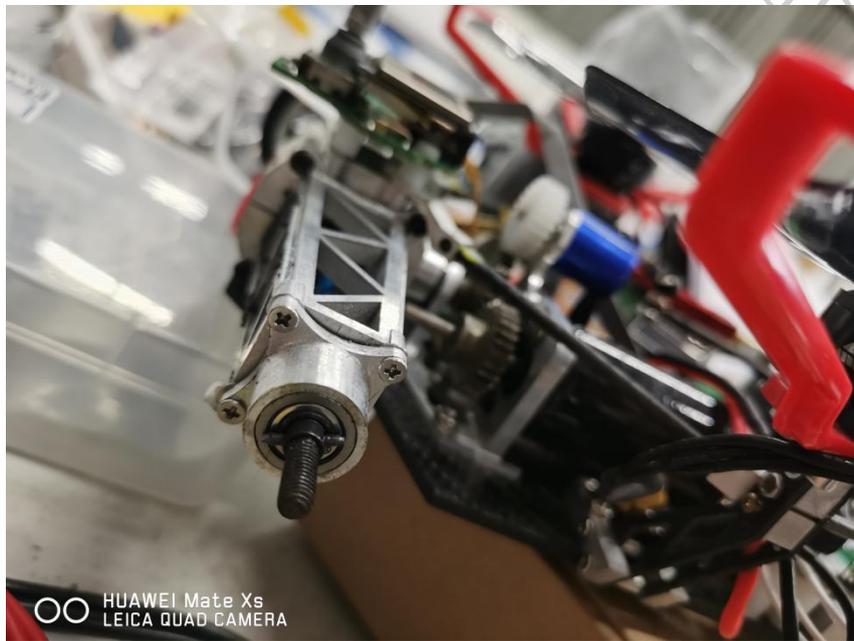


图 1-13：后轮胎接杯输出轴

万向节接杯处使用两组轴承相对压紧的设计，在内侧使用法兰轴承，安装轮胎的过程中利用螺丝对接合器的压力为两侧轴承提供一定预紧力，抵消轴向和径向间隙。不过要注意这并不是很好的设计思路，普通滚珠轴承主要承载径向载荷，对于轴向载荷的承载能力较弱，而且会增加磨损降低寿命，在此使用角接触轴承更好。



图 1-14：万向节安装方式

## 2 硬件设计

该车模的硬件电路由七块电路板组成，按照功能可以分为传感、主控、功率、调参四大部分。在硬件设计中采用了功率和主控分板隔离的设计，将有刷驱动、DCDC 舵机供电、主控供电集中在一块电源板上，双路无刷驱动集成在另一块功率板，两块电路板叠板设计，为了减小无刷驱动等高功率部分对主控模拟信号的影响，将电磁运放和主控芯片设计在另一块电路板放置于远离无刷电机及驱动的部分，板间通过自制的三端子 FC 排线连接。

在以下原理图中，芯片周边阻容配置不一定全部按照器件标识所示，部分阻容我会在焊接时候进行经验性的调整，因此原理图仅作为方案参考。

### 2.1 前瞻 PCB&TOF 测距板

前瞻部分采用八电感循迹方案，三水平一垂直，完成循迹功能，四斜电感完成元素识别。在此基础上，设计了 TOF 测距插板，直接垂直焊接在前瞻 PCB 背板上，TOF 接线通过焊接方式转接到前瞻 FPC 排线上。

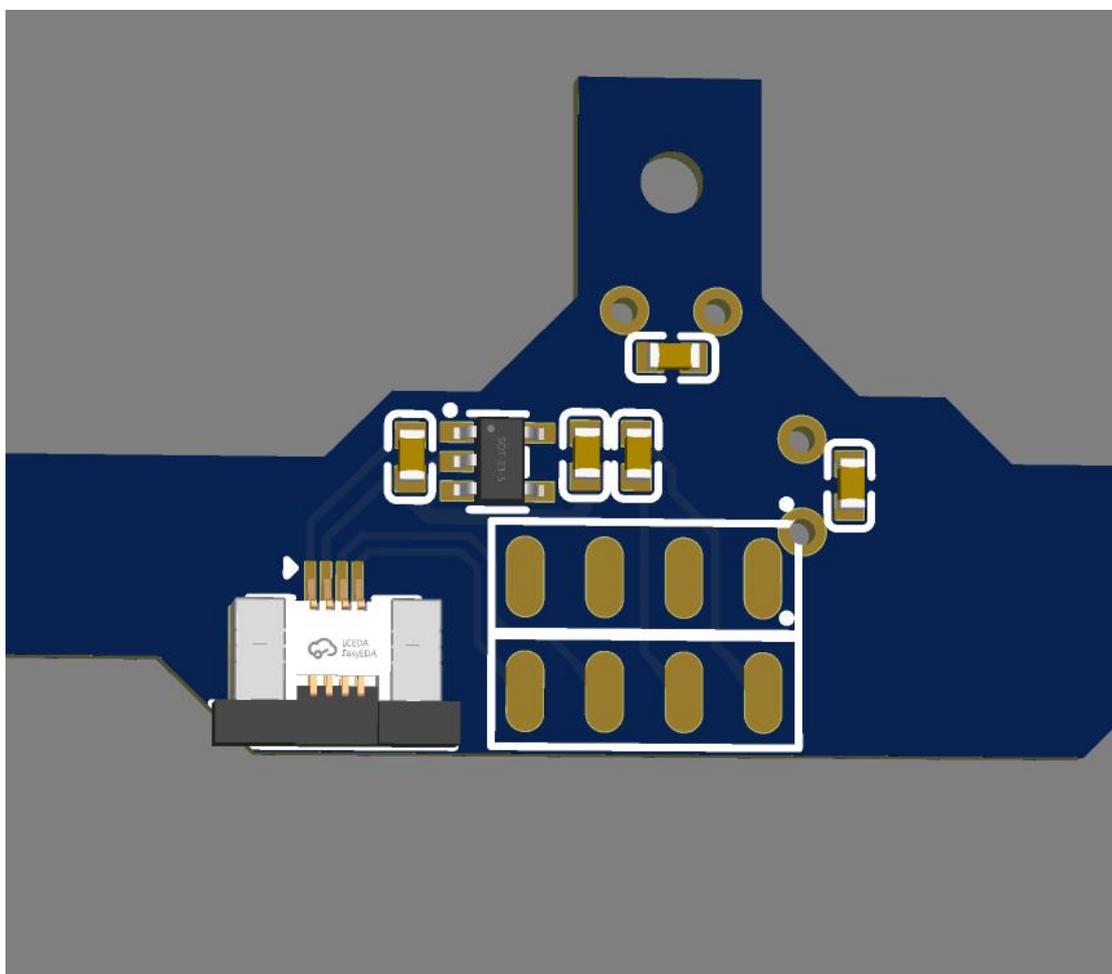


图 2-1：前瞻 3D 图

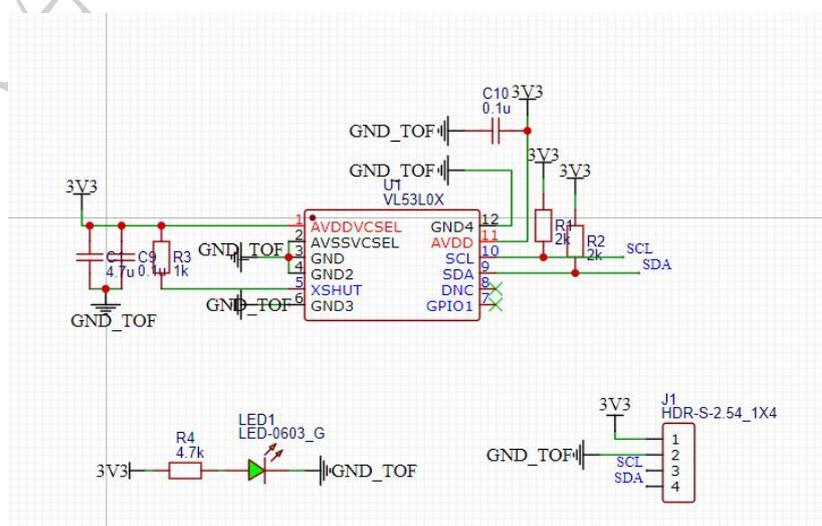


图 2-2: TOF 电路图

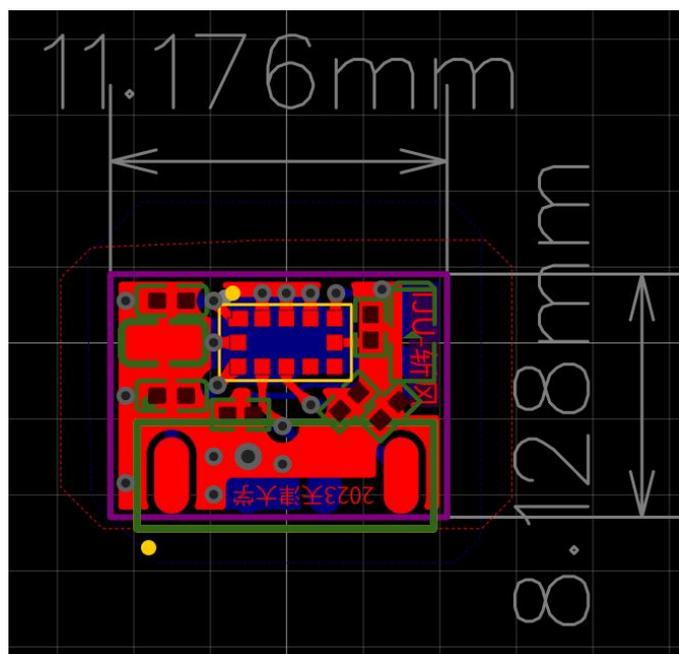


图 2-3: 测距插板 top layer

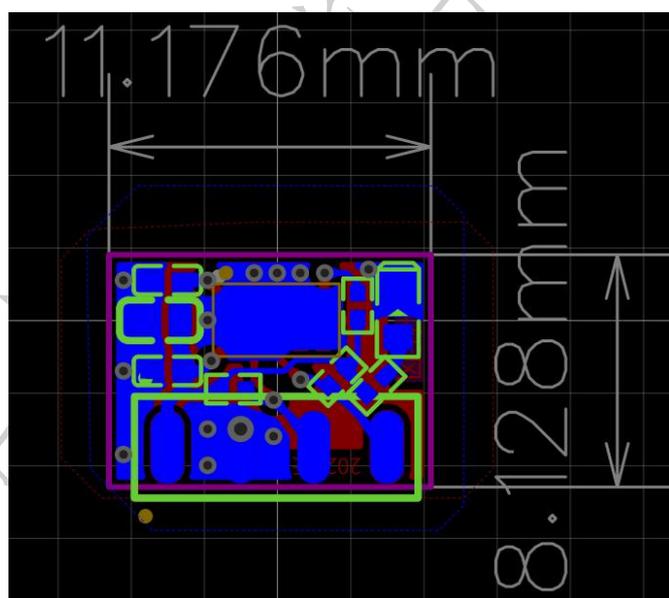


图 2-4: 测距插板 bottom layer

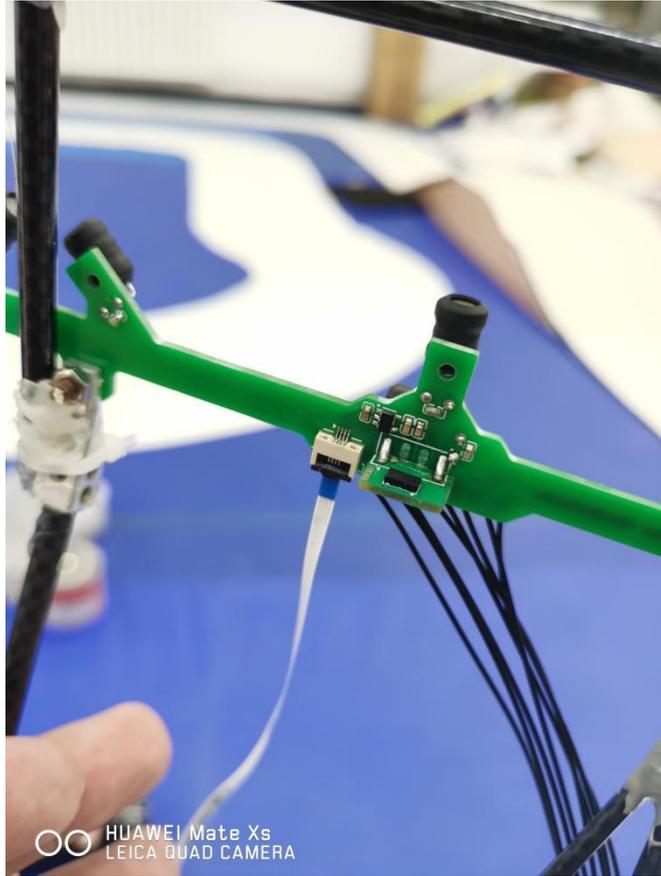


图 2-5：测距插板安装

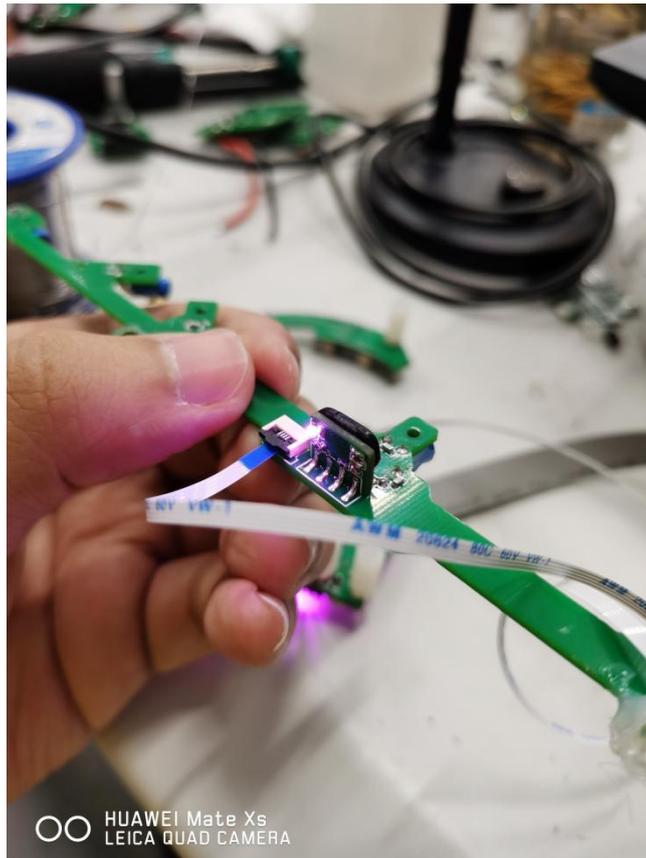


图 2-6：测距插板安装

## 2.2 有刷驱动&功率部分

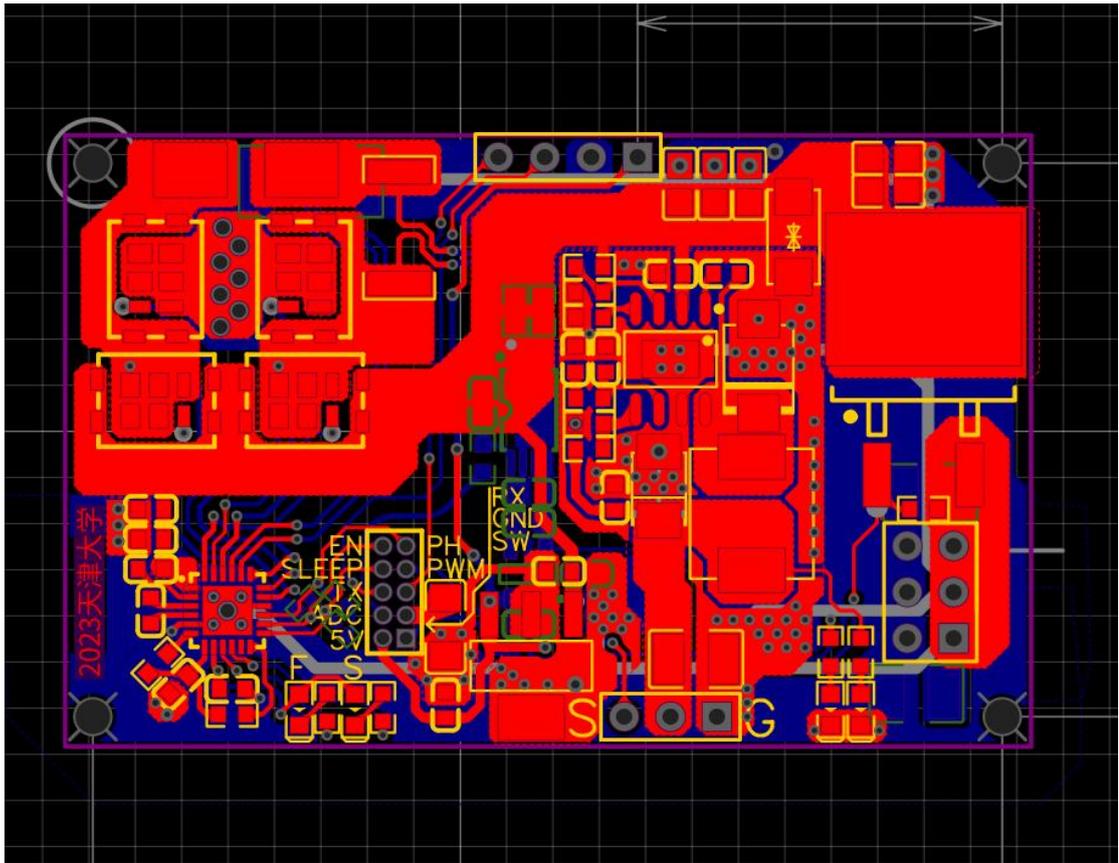


图 2-7：功率板 layout

### 2.1.1 有刷电机驱动

经典 DRV8701 栅极驱动方案，在此不过多赘述。

这里使用 DirectFET 封装的 IRF7480M 作为 H 桥 MOS 管，40V217A，热阻小散热快，注意使用这种封装的 MOS 时候焊接需要注意中心焊盘挤压产生的锡珠影响，可能会导致短路，焊接时候多拿镊子夹起来查看即可。以及调节 IDrive 以适配 7480M 较大的  $Q_g$ 。

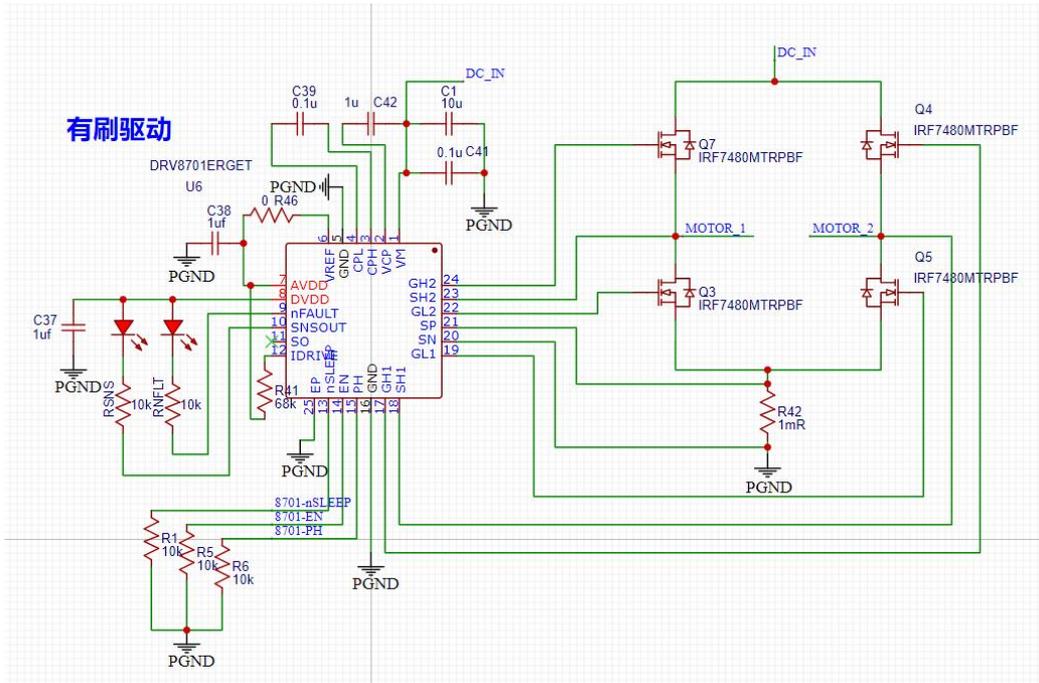


图 2-8: DRV8701

### 2.1.2 舵机供电

在舵机方案中，我们使用了 8.4V 高压舵机，其堵转电流在 3.5A 左右，因此在这里选用了 TPS54560 作为舵机供电，60V5A，支持补偿网络接口和开关频率调节，这里设置反馈电阻使其输出电压为较为激进的 8.5V 左右，在该电压下舵机响应速度为  $0.04s/60^\circ$ ，同时 TPS54560 支持补偿器的接入，在此使用了二型补偿网络， $C_{zero}=4700pF$ ， $C_{pole}=39pF$ ， $R_{comp}=16.9k$ ，获得相对较佳的动态响应。并且提高了 RT/CLK 调节电阻设置的频率为 800kHz 左右，使得输出纹波进一步降低。电路图如下所示。

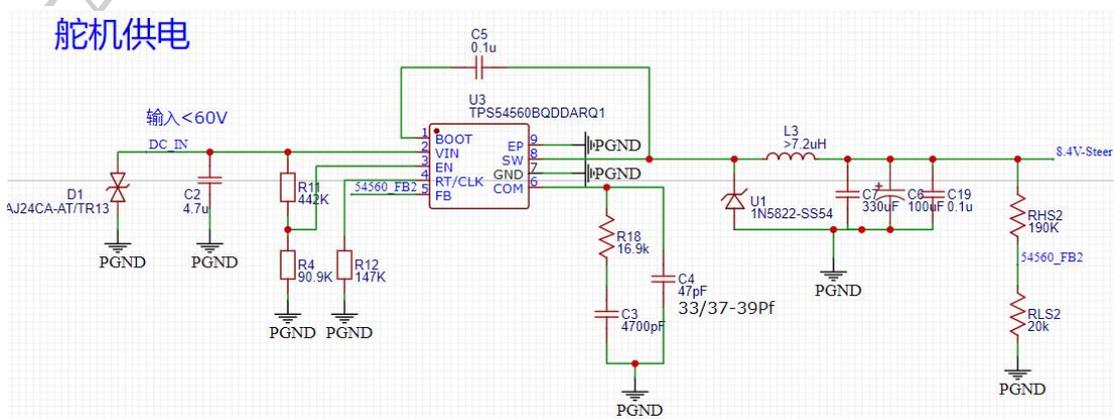


图 2-9: 舵机供电

在 layout 中使用紧凑布局原则，使得 buck 电路的两大工作环路面积相对较小，降低寄生参数，电源工作稳定性提高，对外干扰降低。在背面保留完整地层，同时分割板上模块回流路径，将板内串扰控制在相对较小的范围内。

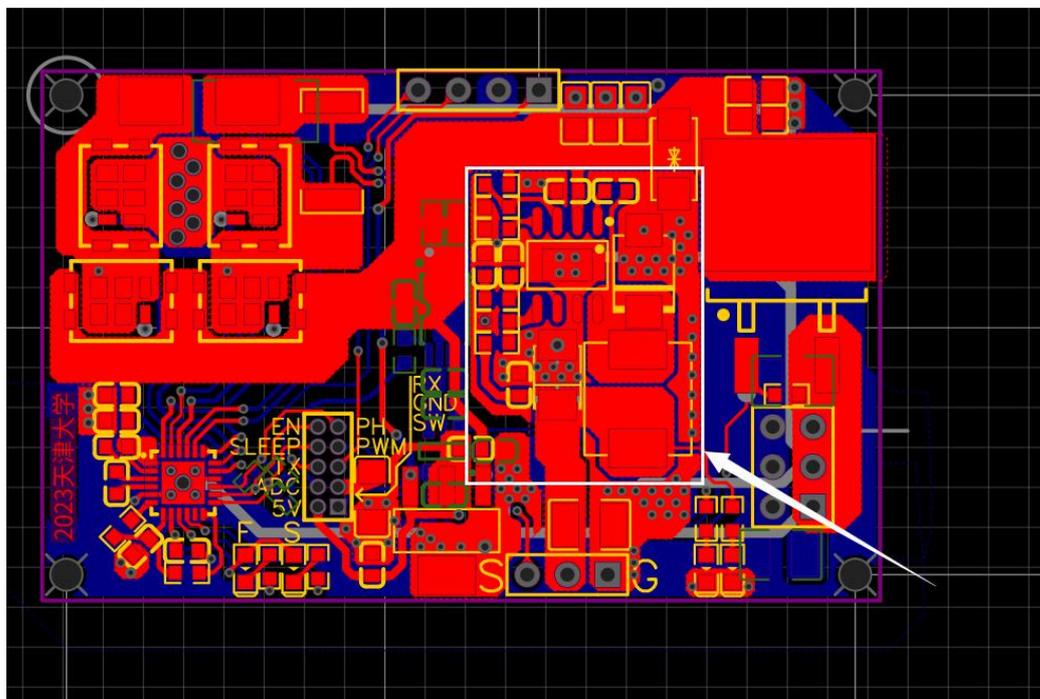


图 2-10: 布局 top

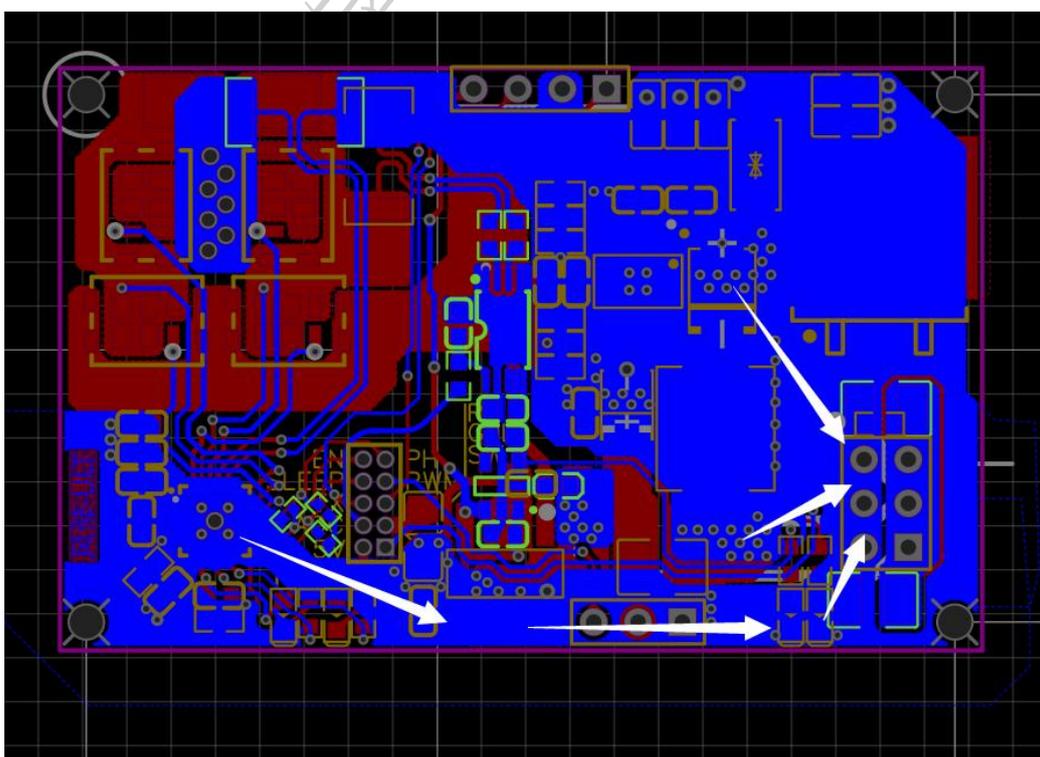


图 2-11: 布局 bottom

### 2.1.3 电流采样

在有刷电机的控制中，底层为电流控制，外层嵌套速度环，虽然我们使用的栅极驱动 DRV8701 内置电流运放，可以在 H 桥下端进行采样，该种采样方式对运放耐压要求较低，只需工作在地电压附近，但缺点是只能采得单向电流，并且对于采样时机有要求：只有下管开启，电流从电机流向 GND 时才可以采到，在下管关闭时采样是失效的（除非在急刹等情况下电机泵升电压对母线发电，走体二极管从地端抽取电流，才会采到反向电流，但这种情况不是常见工况），常见的解决方案为在下管开启时采用触发的方式获得电流值，该方法对于采样时机要求较高，并且在 STC32G 上相对不容易实现，一般做无刷时候用中心对齐 PWM 模式会比较方便地使用注入组等触发方式。因此我们使用了相采样，将采样电阻直接串联在电机相线上，由于电机电流不能突变，因此在任意时刻采到的电流均可用于电流环运算。但是相采样的缺点是采样得到的信号质量相对较差，分辨率低，并且对于运放共模电压要求较高。因此在采样电路设计中我们使用了 INA240A2 作为运放，其具有 50V/V 电压增益，可以在 -4 至 80V 的高共模电压条件下工作，以及具有增强 PWM 抑制功能，可以对驱动 PWM 产生的较大共模瞬变提供高水平的抑制，不会使输出电压产生较大瞬变及相应的恢复纹波，将其静态工作点偏置在 1.65V，即供电电源轨的一半，配合 0.5mR 采样电阻，可以实现 ±60A 左右的电流测量范围，满足所选 480 电机的需要。

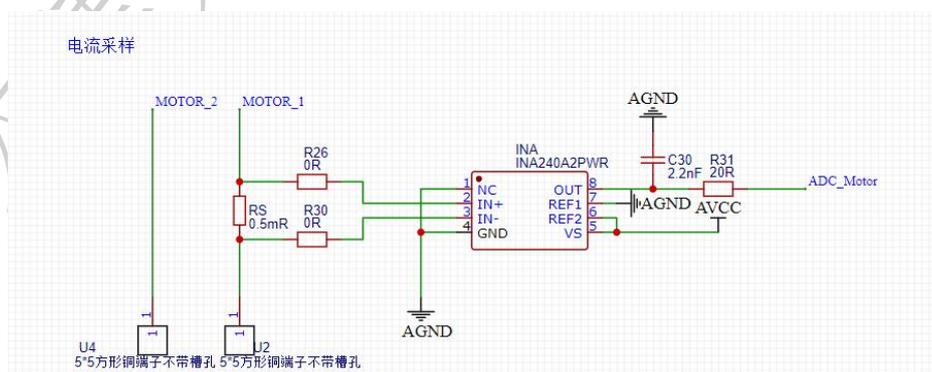


图 2-12: 电流运放

这里需注意采样电阻的两根线需要走差分，遵循等间距并且相对较短原则。一般情况下，即便是使用下桥臂采样也不应将接地的采样电阻的 IN-端直接随意

接入 PGND 网络(不使用 Net-tie)，仍需按照差分原则走线。

#### 2.1.4 LDO

使用 LP5907MFX 作为电流运放供电,其具有高达 82dB 的 PSRR,并且支持 m1cc 作为输入输出电容,获得高精度高稳定性的运放电源。

主控板的 5V 供电同样放置在了该功率板上,使用一颗 7805 直接将 12V 降压到 5V,通过排线送到主控板上,在设计初考虑不将 12V 引入到主板上,但这并不是一个很好的设计,一是压降过大,二是增加了电源面积,好在主控工作电流并不大,并且有负压风扇提供的散热,可以使得 7805 不会过热关断。

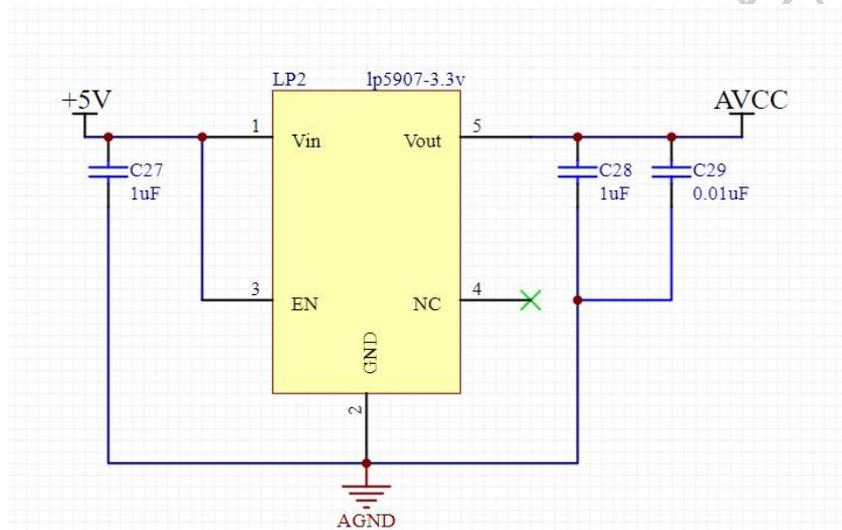


图 2-13: LDO

#### 2.1.5 遥控开关设计

在功率板的设计中,考虑到增加遥控关机的需求,使用一 PMOS BBS3002(-60V,-100A)控制功率部分的 12V 供电,PMOS 的栅极通过信号排线接在调参板上的可拆卸遥控模块上,实现遥控整车电源的关闭,提高调试时的安全性,在正式比赛时拔掉遥控模块,换用跳线帽即可接通电源。

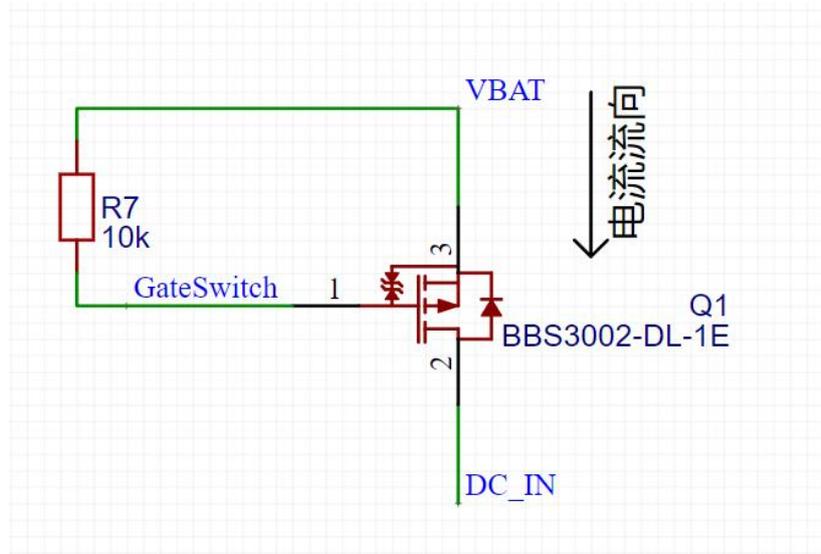


图 2-14: PMOS 开关

天津大学智能车队

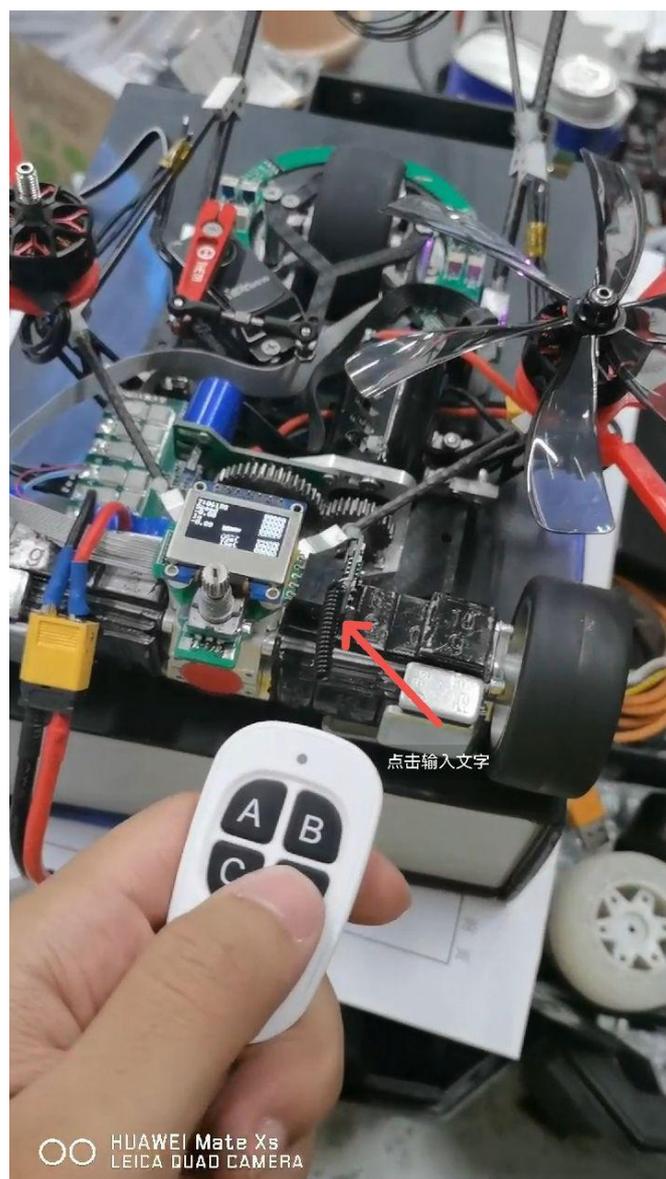


图 2-15：遥控控制

注意在这里的设计中遥控模块会把自身的电源也断掉，因此只能作关机使用，因此在遥控模块上焊接一按键，按下按键整车上电，再按下遥控开机，即可保持常开状态，并实现双保护。

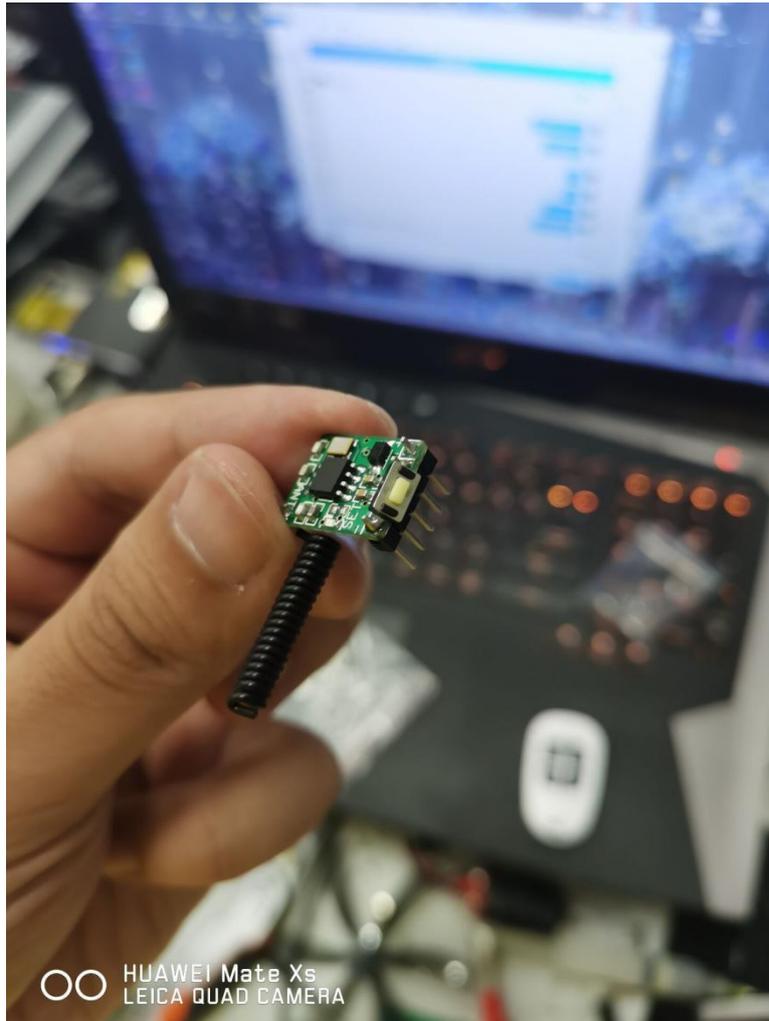


图 2-16：遥控模块

### 2.3 无刷驱动

无刷驱动采用 STC15W408AS+EG2133 方案，由主控芯片、过零检测电路、三相半桥组成。

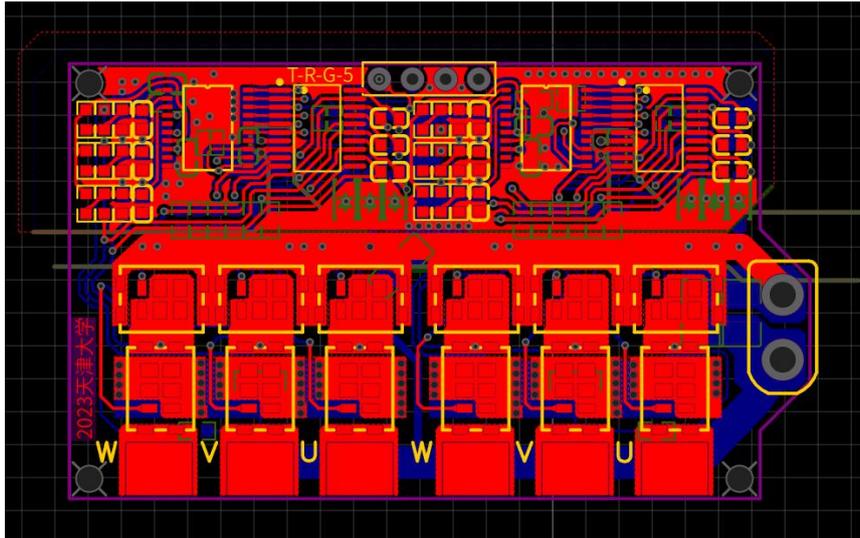


图 2-17：无刷驱动 layout

这里将两个单片机串口挂在同一个串口总线上，添加 0402 跳线电阻后下载时通过拆焊电阻来分别对两个单片机进行下载。车模主控芯片给驱动发送速度数据也是通过该串口进行发送。

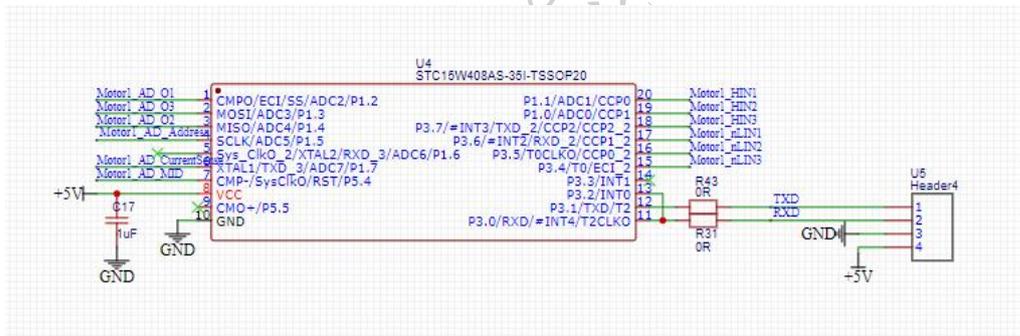


图 2-18：无刷驱动主控

过零检测采用经典的反电动势分压采集电路，Motor\_VS1/2/3 为三根相线接入，Motor\_AD\_01/2/3 为分压输出，Motor\_AD\_MID 为估测的变形后的中性点电压输出，四个输出引脚接入单片机的模拟比较器，使用两相通电的驱动策略，每次对悬空相分压输出和中性点电压作比较，即可检测出各相的过零事件。

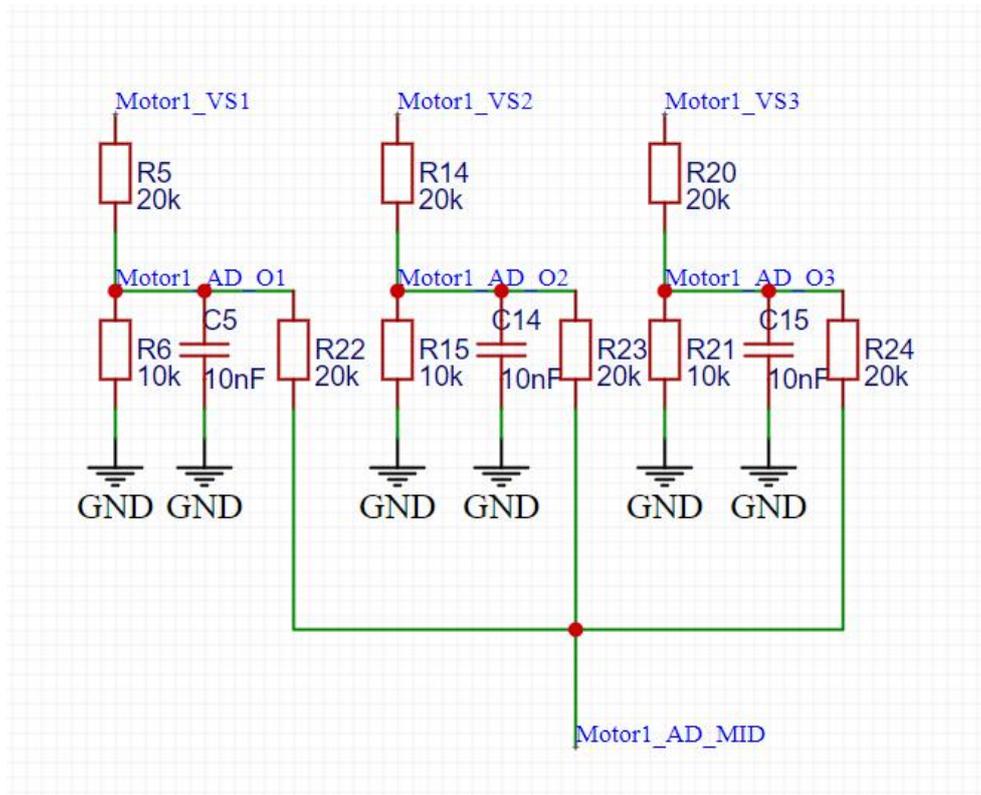
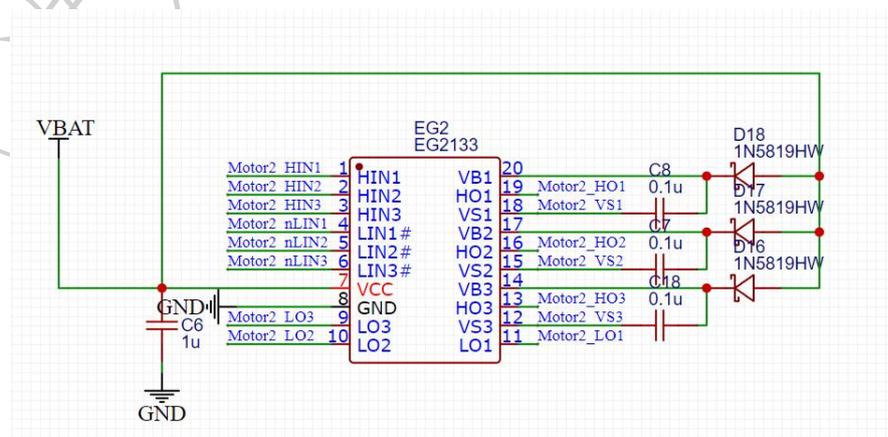


图 2-19：过零检测电路

半桥驱动使用 EG2133，HIN、LIN 同相输入，HO、LO 反相输出，在这里只需要三组 PWM 即可完成六个 MOS 的控制。注意其耐压值最高为 25V，使用 H 桥母线电压进行供电时需要增加 TVS 保护，防止电机泵升电压冲坏驱动。

MOS 同样采用 IRF7480M；





## 2.4 主控&运放一体板

主控芯片采用 STC32G12K128, 引出 ADC, SPI, IIC, UART, PWM 等片上外设, 在这里不多赘述。

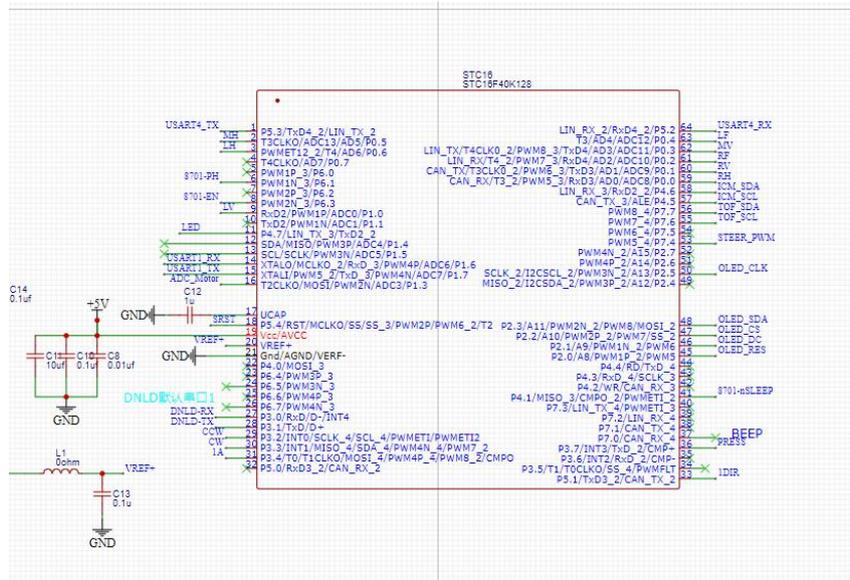


图 2-22: 主控 MCU

运放使用两片 OPA4377 完成电磁信号的放大输入。运放供电同样使用 LP5907。

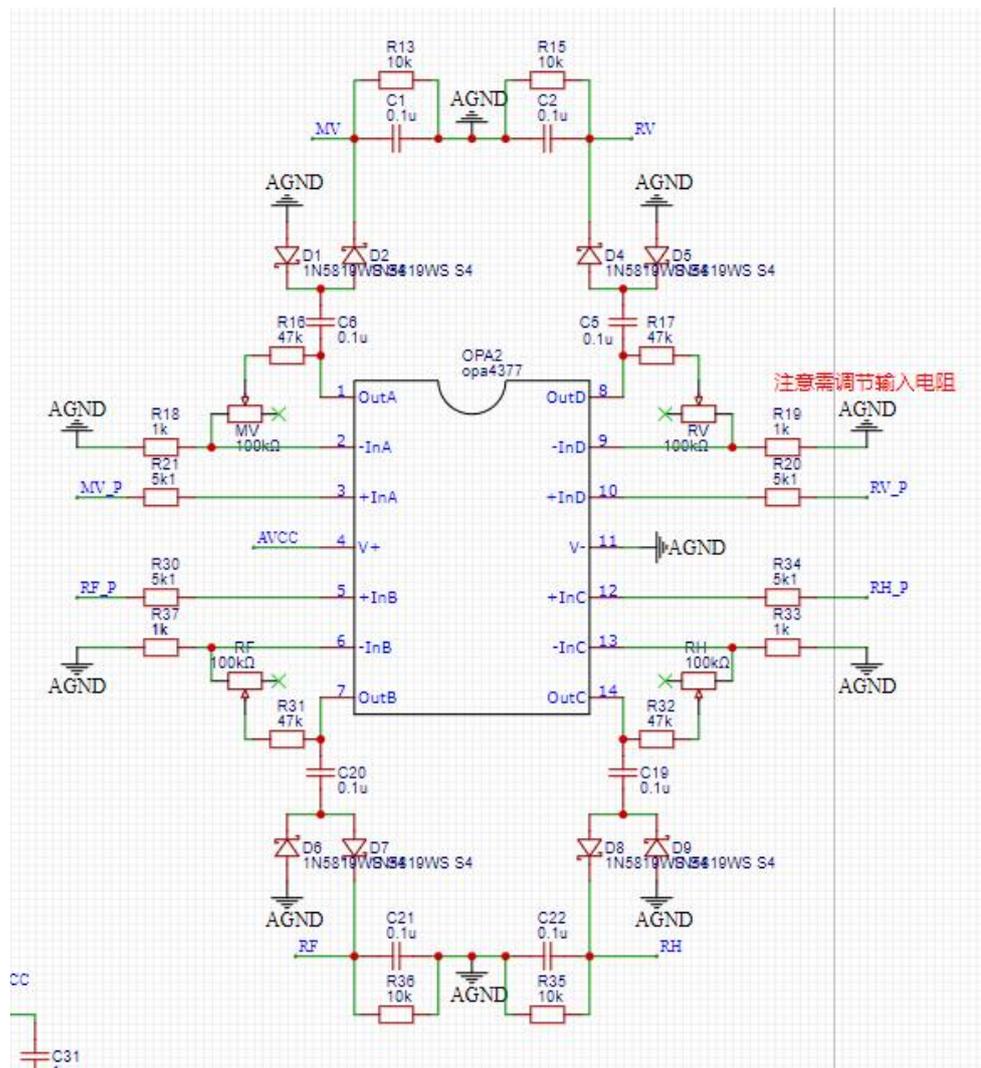
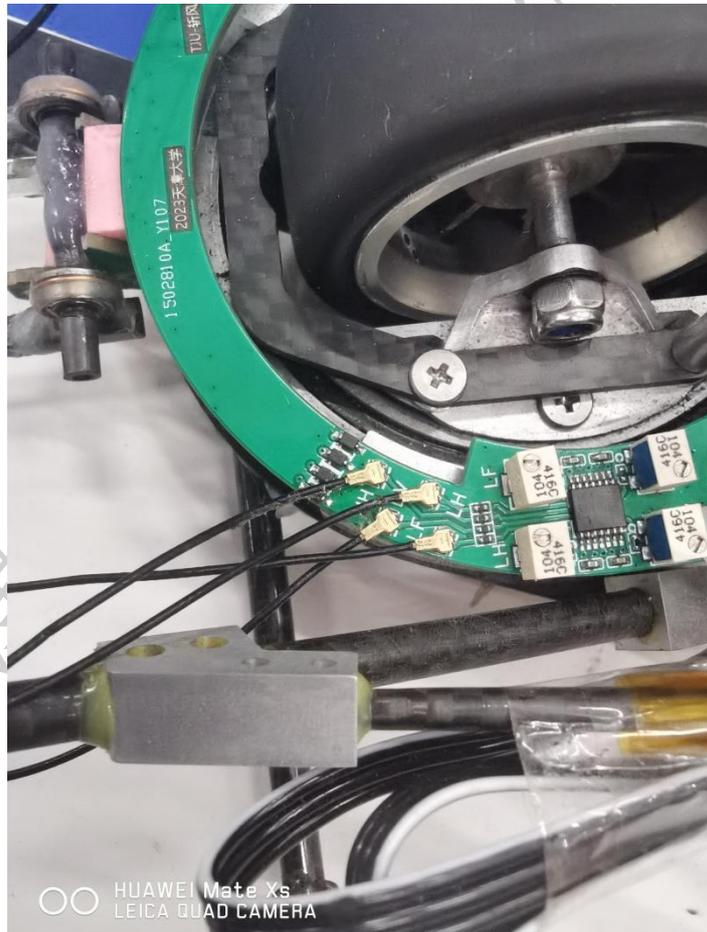
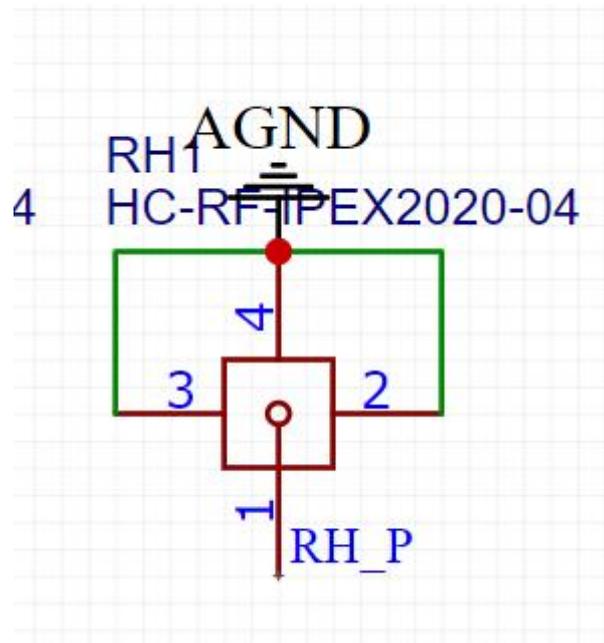


图 2-23: OPA4377

运放输入与前瞻的连接采用了 0.81mm 同轴线，外层为 GND，内层为信号，信号质量和抗干扰性较好。



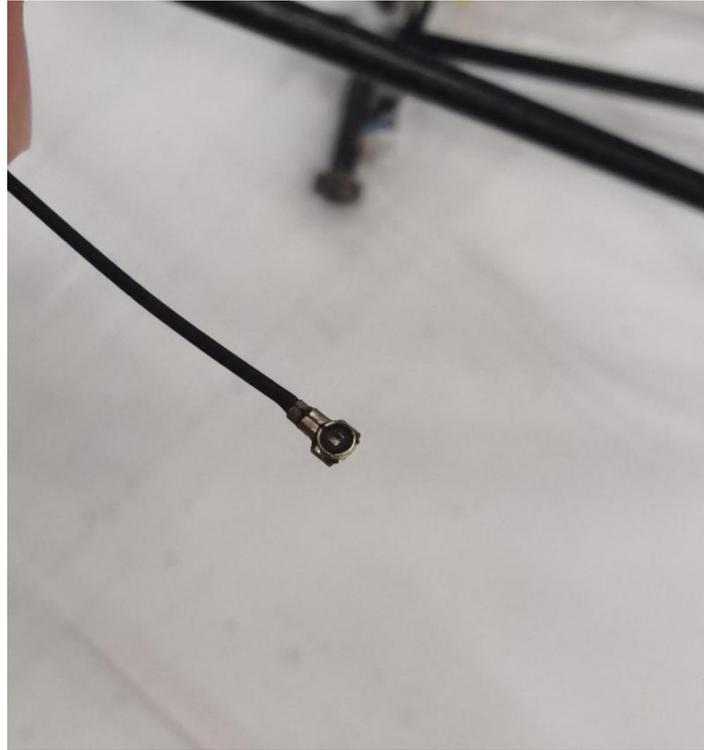


图 2-24：同轴线方案

在最初的设计中，将主控运放和有刷驱动板设计为叠板结构，放置于整车的左侧空间。

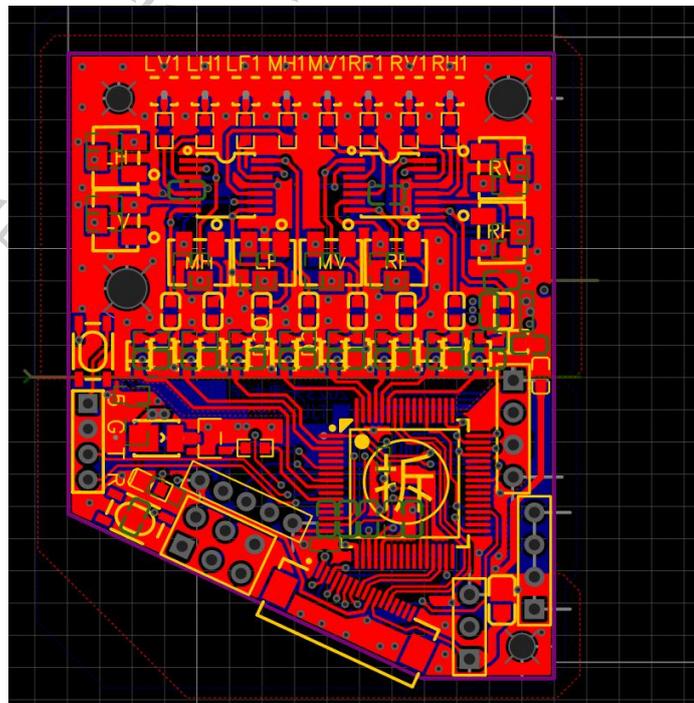


图 2-25：初版 PCB

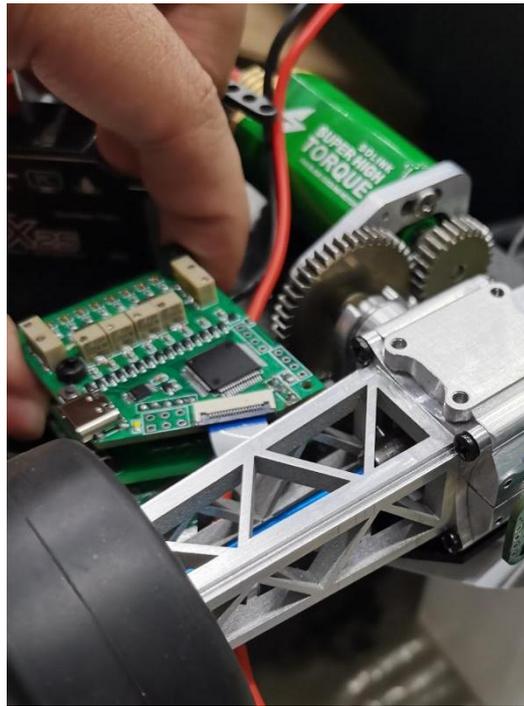


图 2-26：初版 PCB

但是在前期的调试中，其拆卸难度较大，维护困难，且难以调节运放倍率，并且会受到电机影响，使用屏蔽布隔离后虽然噪声降低了 70%但是依然存在很大干扰。后续考虑利用车头空间，设计一环形主控板安装在前轮转向总成周围，提高空间利用率，并且远离无刷驱动板，远离有刷电机的径向方向，降低受到的干扰。

在 UG 中先绘制 PCB 草图，确定形状并检查干涉后导出 DXF 到立创 EDA 中作为板框层。

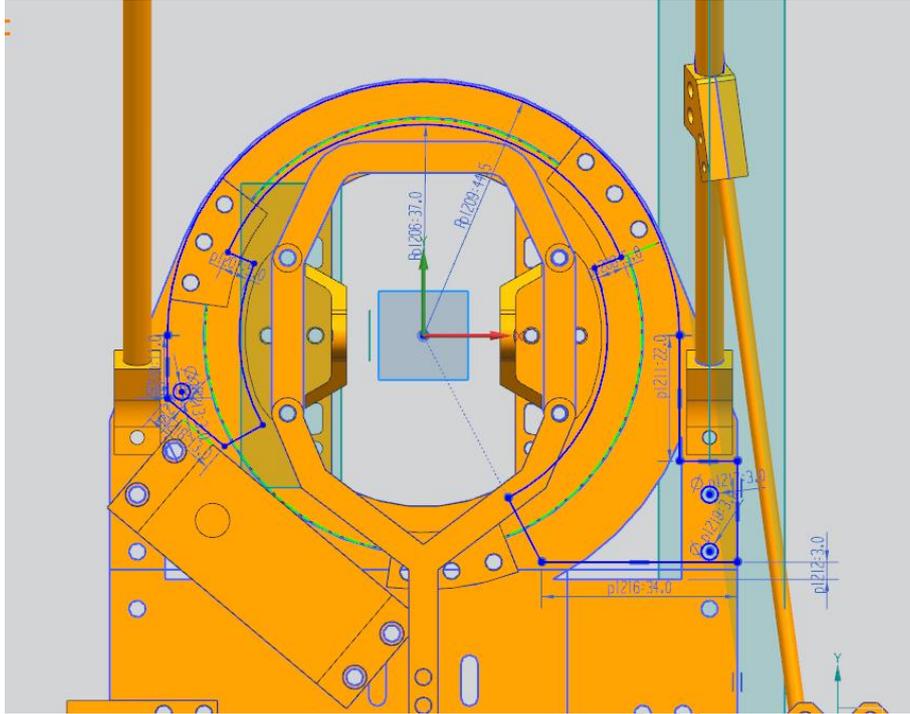
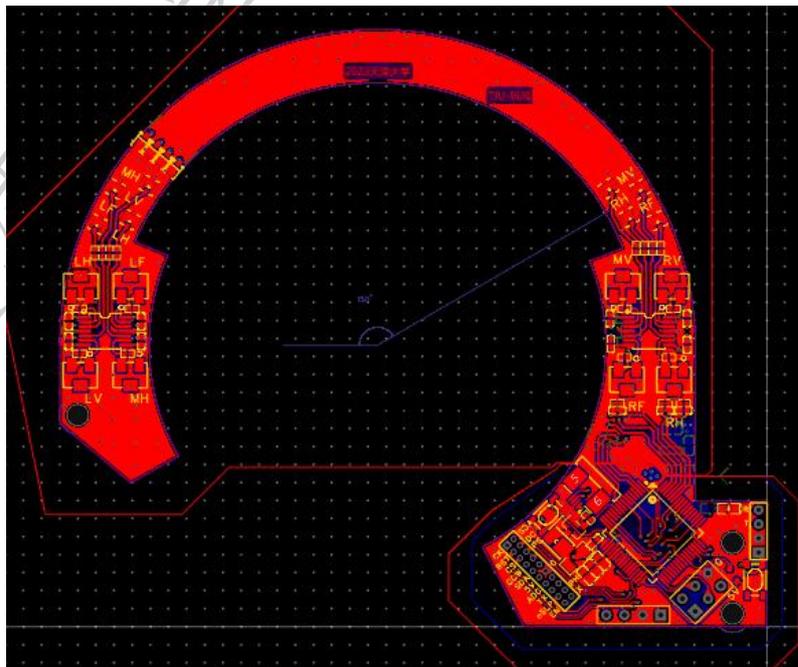


图 2-27：PCB 草图

将两块 OPA4377 分别布局在左侧和右侧空间，使用 3224W 小电位器替代 3296W，以节约空间。



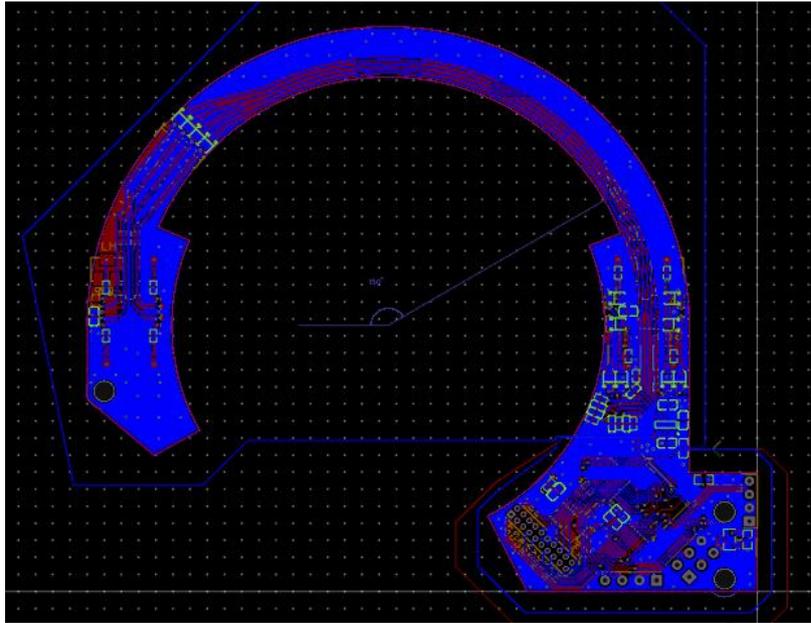


图 2-28：PCB layout

最终安装在车模上。

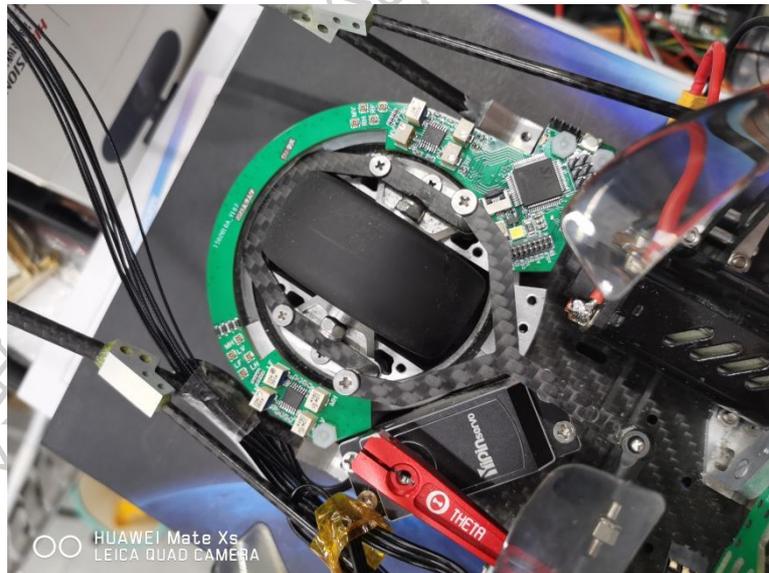
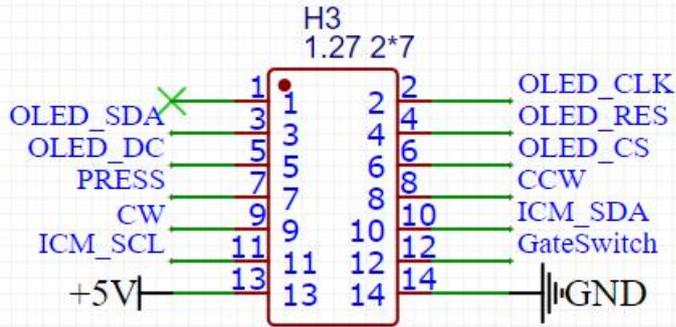
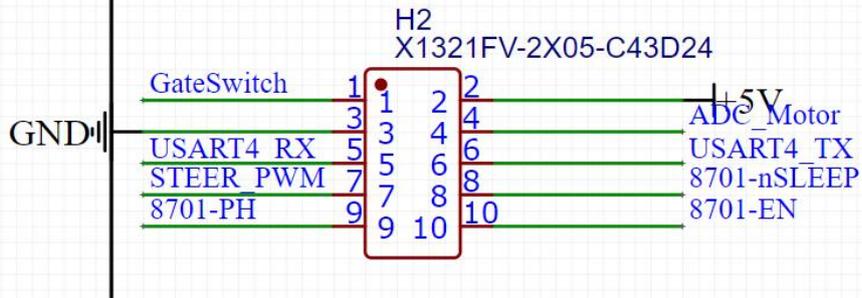
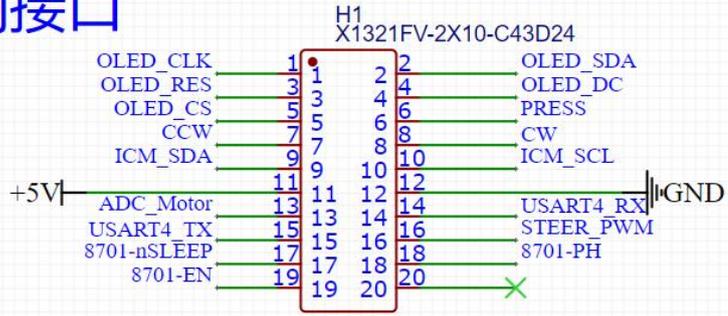


图 2-29：PCB

主板、驱动板、调参板三块 PCB 使用一根信号排线进行连接，通过合理分配接口线序，自制一压接 FC 排线，排线上分布三个穿刺端子，即可对应三块电路板的分布位置。

# 板间接口



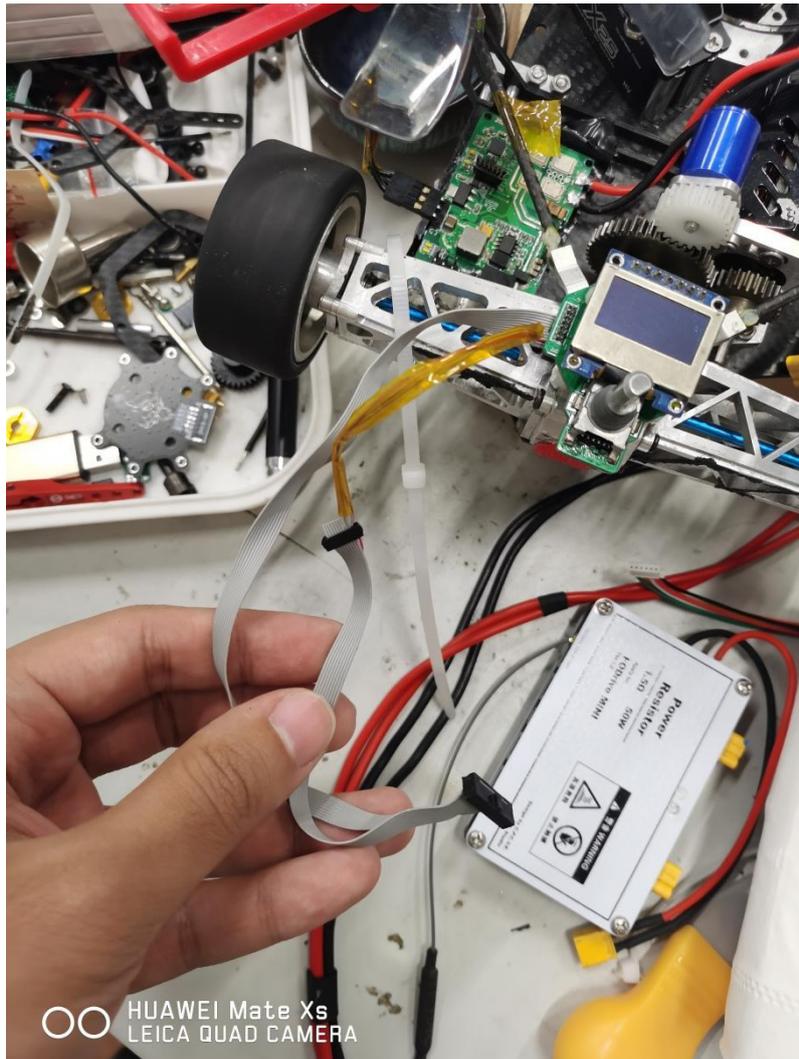


图 2-30：压接三端子排线

## 2.5 调参板

调参板兼具陀螺仪安装板和开关板功能，将调参放置于尾部可以增加安全性，避免手伸进负压风扇里调参的情况发生。调参采用一个 EC11 调参旋钮，配合滚动菜单来实现参数的选择和调节。屏幕放置在调参旋钮前方，底部陀螺仪 ICM20602 放置在屏幕下方。

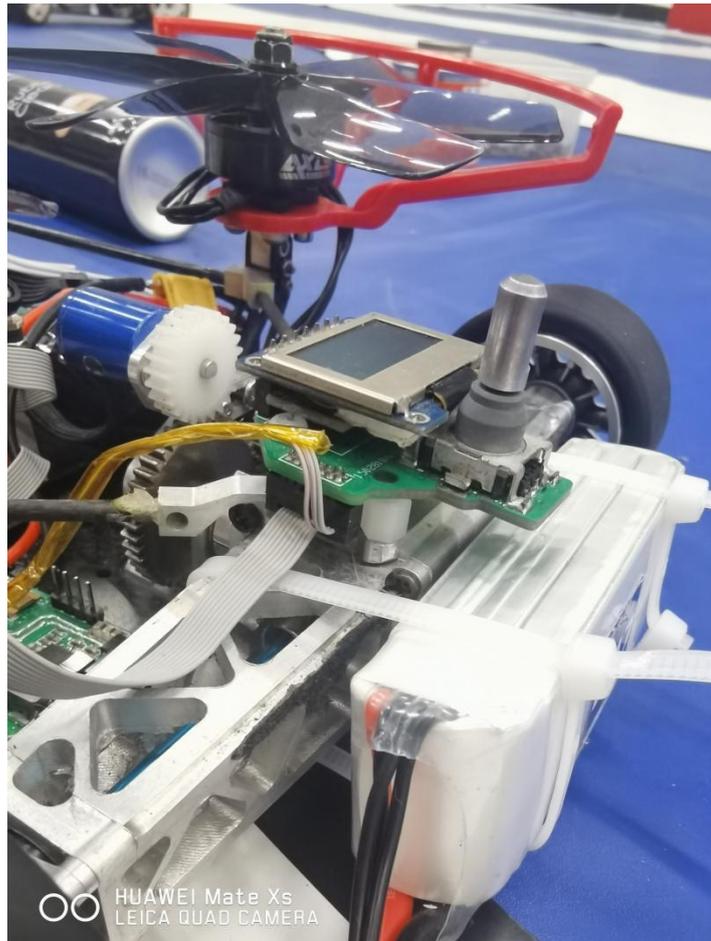


图 2-31: 调参板安装

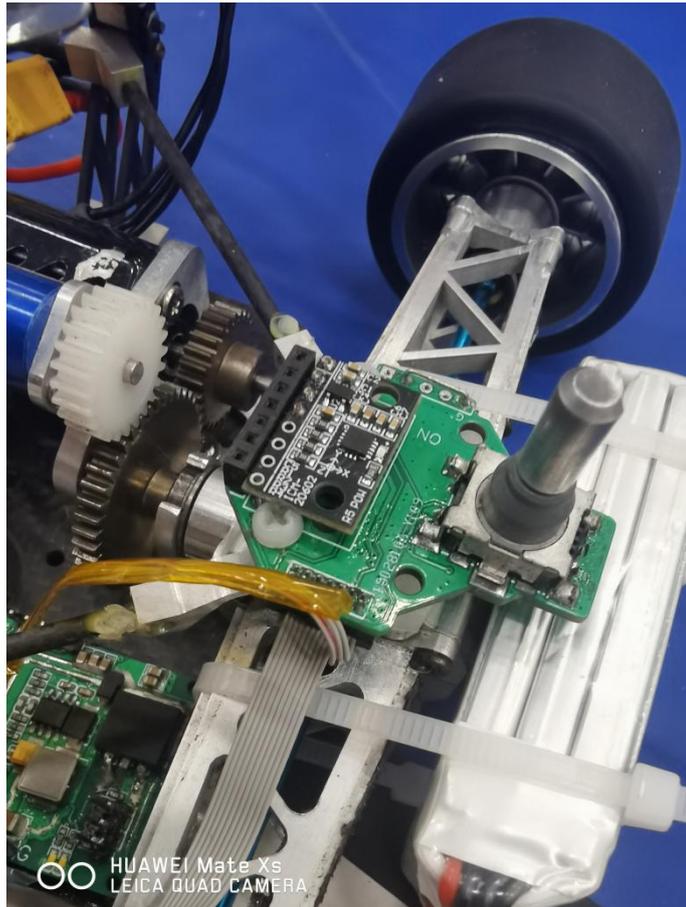


图 2-32：陀螺仪安装

### 3 软件设计

#### 3.1 主控部分程序设计

整车控制周期 2ms，控制流程为读传感器(陀螺仪、编码器、电感)，电流采样，电流控制，速度控制，标志位状态机，循迹算法，输出舵机电机控制信号。

电磁循迹的算法就是经典的差比和算法，各年度电磁组别技术报告中均有提及，在此不过多赘述。

在速度控制中，速度环内嵌电流环，速度控制器的输出作为电流控制器的输入，电流控制再输出给电机 PWM。

在使用速度环单环控制电机时(即直接进行电压控制)，可以保证闭环的效果。电机保持稳定转速的条件是转矩与外界阻力平衡，当该平衡一旦被打破就会引起速度的变化，进而引起电流变化，电流决定电机扭矩。而电机及其所带负载系统是有惯性的，在偏离设定速度的情况下，无论施加多大电压都不会直接将转速拉到设定值上，而施加额外电压带来的校正扭矩越大，电机的加速度也会越大，越

容易出现过冲或震荡。并且在电机速度趋近于饱和的情况下，反电动势同样会升高，此时单纯的电压控制会逐渐脱离线性，电机响应降低，在增加电流内环后，可以更好的对转矩进行控制，保证了整体的控制效果。

```

157 void TM4_Isr() interrupt 20
158 {
159     →T+=PIT_ms;
160
161     //1. 读传感器
162     →GetPeriphData();
163     //2. 电流环
164     →MotorCurrentRegulating();
165     //3. 运行状态控制
166     →status_select();
167     //4. 循迹、元素识别
168     →data_process();
169     //5. 输出到外设
170     →data_output();
171
172
173     →TIM4_CLEAR_FLAG; //清除中断标志
174 }

```

图 3-1: 主中断

```

137 /**
138  ** Function name: GetPeriphData
139  ** Descriptions: 读取传感器数值
140  ** Input parameters: None
141  ** Output parameters: None
142  ** Returned value: None
143  ** Remarks: 电流读取不包含在内
144  ***/
145 void GetPeriphData()
146 {
147     GetGyroscopeData(&indata); //获取加速度
148     GetEncoderSpeedData(&indata); //获取车辆速度
149     PeriphDataReconstruction(&indata, &flag, &setpara); //角度、速度重构
150     →get_ad_newvalue(); //获取前瞻8电感值
151 }
152

```

图 3-2: 传感器读取

```

123 | *****
124 | ** Function name: MotorCurrentRegulating
125 | ** Descriptions: 电流环
126 | ** Input parameters: None
127 | ** Output parameters: None
128 | ** Returned value: None
129 | ** Remarks: 包含速度环外环, 电流环内环
130 | *****/
131 | void MotorCurrentRegulating(void)
132 | {
133 |     MotorADCStartOnce(); // ADC采样一次, 获取电流初始AD值
134 |     MotorADCValueStorage(); // 偏置并存储电流AD值
135 |     MotorPreCurrentFilter(&indata); // 滑动平均滤波
136 |     MotorCurrentReconstruction(&indata); // 电流重构为真实安培数
137 |     MotorSpeedPIDControl(&indata, &setpara); // 速度环PID
138 |     MotorCurrentPIDControl(&outdata, &indata, &setpara); // 电流环PID
139 | }
140 |

```

图 3-3: 电机控制

```

89 | *****/
90 | #if 1
91 | void MotorCurrentPIDControl(struct outdata_s *poutdata, struct indata_struct *pindata, struct setpara_s *psetpara)
92 | {
93 |     // 获取实际值
94 |     pindata->MotorCurrentPID.pre = (float)pindata->Motor_Current_Value;
95 |     // 获取目标值
96 |     pindata->MotorCurrentPID.tar = (float)psetpara->SetMotorCurrent10X / 100;
97 |     pindata->MotorCurrentPID.tar = (float)pindata->MotorSpeedPID.out;
98 |     // 计算偏差
99 |     pindata->MotorCurrentPID.bias = pindata->MotorCurrentPID.tar - pindata->MotorCurrentPID.pre;
100 |     // 计算PID输出值, 1是比例系数
101 |     pindata->MotorCurrentPID.out += ((float)psetpara->CurrentPID_Kp * 1.0f) * (pindata->MotorCurrentPID.bias - pindata->MotorCurrentPID.lastBias) \
102 |     + ((float)psetpara->CurrentPID_Ki * 1.0f) * pindata->MotorCurrentPID.bias;
103 |     // 保存偏差
104 |     pindata->MotorCurrentPID.lastBias = pindata->MotorCurrentPID.bias;
105 |
106 |     // 输出限幅
107 |     if (pindata->MotorCurrentPID.out > (float)psetpara->CurrentOutMax)
108 |     {
109 |         pindata->MotorCurrentPID.out = (float)psetpara->CurrentOutMax;
110 |     }
111 |     if (pindata->MotorCurrentPID.out < (float)-psetpara->CurrentOutMax)
112 |     {
113 |         pindata->MotorCurrentPID.out = (float)-psetpara->CurrentOutMax;
114 |     }
115 |     DynamicCurrentLimit();
116 |
117 |     poutdata->MotorSpeedPWM = pindata->MotorCurrentPID.out; // DEBUG for Current PID Circ
118 |
119 |
120 |
121 | #endif
122 |

```

图 3-4: 电流环 PID

在内环电流环参数精确整定之后，速度环的参数就不那么敏感，粗略调校即可达到相对较好的控制效果。注意调节电流环的时候依旧需要把车放在地上调，空载状态下只会较低速度上有较好的控制效果，而转速拉得比较高仍然会使得电流控制失效。

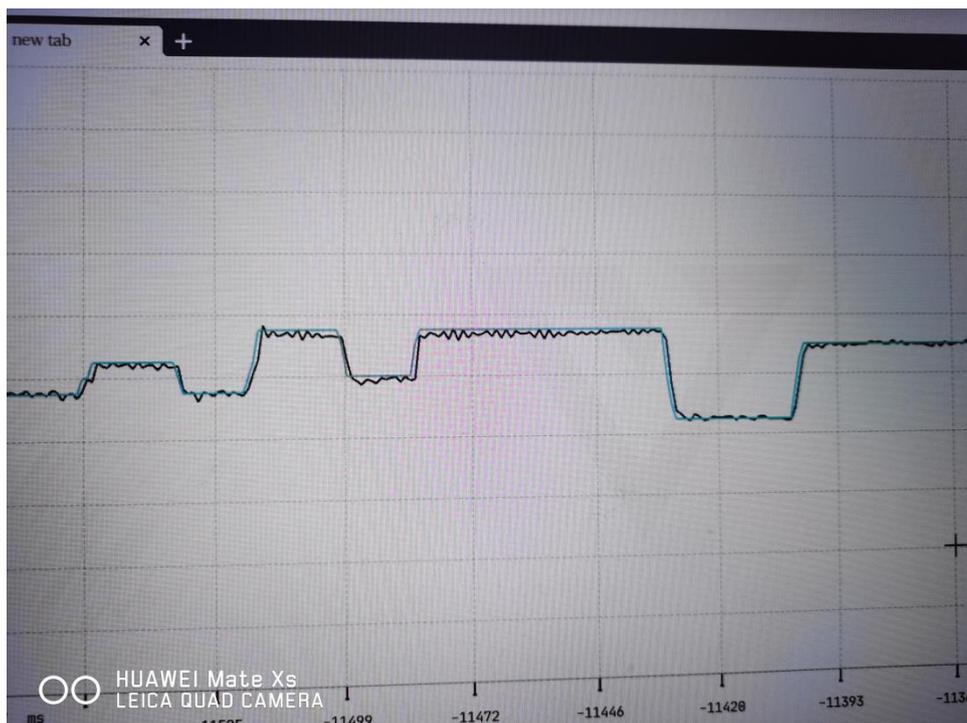


图 3-5：速度控制曲线

在传统车模运行的过程中，由于速度环参数调校得不适宜，以及车模阻力变化的影响，电机的负载转矩常常会随机突变，在此情况下驱动系统往往会频繁对速度进行较为高频的调节，容易引起电流震荡，影响转矩输出和速度稳定性，加剧电机发热。因此在这里使用了动态电流限幅控制，在电流控制器的输出增加一可以控制限幅阈值的PI 控制器，该阈值控制器的输入端来自速度环的输入端，并根据车模状态机动态切换其负增益，在出库、坡道等需要电机短时间高过载运行时解除对电流的动态限制，而在无元素运行时恢复控制，使得电机电流及扭矩变化更加柔和。使用该种控制策略，车模在数小时的间隔运行后电机温度仅会略高于常温，不会出现过热现象，有效降低了电机的损耗和高温退磁现象。

### 3.2 无刷电调程序设计

负压风扇使用的是无感 BLDC，相比于有刷电机的机械换相(换相点和转子位置物理绑定)，无刷电机要想获得旋转的同步磁场，必须获得转子位置，在无感 FOC 控制中可以使用多种观测器或注入波形的方法获取转子角度，但是对于 STC 单片机来说，算力要求较低的反电动势六步换相算法更加适宜。这里使用两相通

电一相悬浮的驱动方法，通过检测悬空相的感生电动势来测量转子位置。

如下图举例，在 AB 相通电期间，可以看出 C 相切割磁感线产生的反电动势方向由正变负，因此如果可以测得 C 相线圈两端电压，可以发现它在整个 AB 相通电期间会整个换一个方向，也就是过零点。

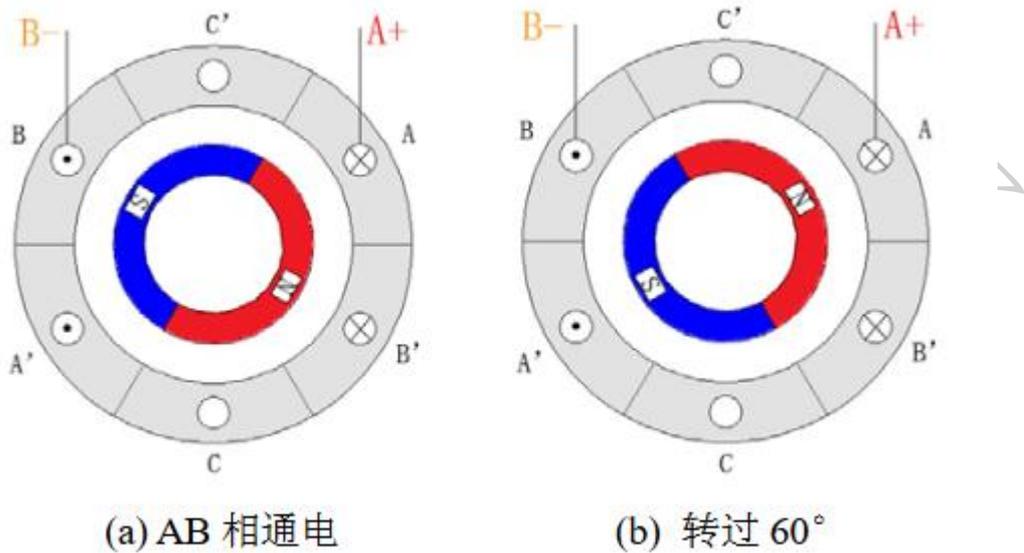


图 3-6：转子运行图示

这也就是上文硬件过零检测电路的作用，由于电机绕组接线方式的原因，无法直接测得中性点电压，因此通过电阻网络获得一个虚拟中性点，即可完成对悬空相两端的反电动势检测。六步换向中一个电角度内换相 6 次(三相半桥合成的 6 个扇区所能获得的所有磁场方向)，间隔  $60^\circ$  角，也就是说，在检测到悬空相相线电压等于中性点电压时，即该相反电动势过零时，就证明转子已经转过  $30^\circ$  电角度了，因此利用 STC15 上的比较器模块，在每个扇区内等待比较器中断，进而获得过零点位置。这里注意比较器中断会受到退磁干扰，原因是接入相切到悬浮状态时，由于绕组电感电流不能突变，需要一段时间续流消磁，而且负载电流越大退磁时间相对越长，因此在换相后第一次获得到的过零信号大概率是消磁产生的毛刺，因此等待稳定后再去获得过零点。不过单纯的等待消磁还是有概率漏掉过零事件。

```

241 | *****
242 | ** Function name: cmps_isr
243 | ** Descriptions: 比较器中断
244 | ** Input parameters: None
245 | ** Output parameters: None
246 | ** Returned value: None
247 | ** Remarks: 延迟一段时间后再次进行判断, 滤除干扰
248 | *****/
249 | void cmps_isr(void) __interrupt(21)
250 | {
251 |     uint8_t cmps, i;
252 |     CMPCR1 &= ~BIT(6); //清除CMPPIF
253 |     i = 8; //约延迟10us
254 |     while(i--);
255 |     cmps = CMPCR1 & BIT(0);
256 |     if(CMPCR1 & BIT(5)) //上升沿中断7532
257 |     {
258 |         if(cmps == 0) //若输出为低则退出
259 |         {
260 |             return;
261 |         }
262 |         else //else CMPCR1 & BIT(4) 下降沿中断7432
263 |         {
264 |             if(cmps == 1)
265 |             {
266 |                 return;
267 |             }
268 |         }
269 |     }
270 |     if(MotorState & FLAG_START) //电机已启动条件下
271 |     {
272 |         VectorCountCalculate(); //计算平均换相时间, 延迟换相
273 |         MotorStepRunNext(); //换相函数
274 |     }
275 |     if(MotorZeroCount < 200)
276 |     {
277 |         MotorZeroCount++; //过零点计数
278 |     }
279 | }
280 |
281 |
282 |
283 |
284 |
285 |

```

图 3-7: 比较器中断

检测到过零点位置后, 即可进行换相, 这种方式最为简单粗暴, 但是立刻换相会导致新接入的绕组处于平衡位置, 通电但是不受力, 需要在转子惯性作用下越过平衡点才可以获得转矩, 因此效率会降低, 发热增加, 转矩波动大。因此在检测到过零点后延迟  $30^\circ$  电角度为最佳换相点, 可以在过零检测电路中增加电容, 起到硬件延迟换相信号到来的作用, 但缺点是只能在对应转速下起到较为稳定的延迟效果。另一种方法为软件延迟换相, 在转速变化率不大的情况下, 近似认为转子转速在一个扇区内的速度恒定, 因此通过另一定时器 Tim0 记录从换相切换到检测到过零点的这段时间, 再延迟同样的时间进行换相, 这样就实现了大概  $30^\circ$  的延迟换相。

```

173 | /*****
174 | ** Function name: MotorStepRunNext
175 | ** Descriptions: 换相函数
176 | ** Input parameters: None
177 | ** Output parameters: None
178 | ** Returned value: None
179 | ** Remarks: 使用2000KV左右的无刷电机时，4s供电
180 |           极对数14，电频率拉满约为4.6KHz，>=200us
181 | *****/
182 | void MotorStepRunNext(void)
183 | {
184 |     →static uint8_t step = 0;
185 |     →
186 |     →step++; //步序+1
187 |     →
188 |     →if(step >= 6) //完成一个电角度周期，步序回零
189 |     {
190 |         →step=0;
191 |     }
192 |     →if(MotorState&FLAG_REV == 1) //电机反转
193 |     {
194 |         →switch(step)
195 |         {
196 |             →→→case 0:MOTOR_REVSTEP0;break;
197 |             →→→case 1:MOTOR_REVSTEP1;break;
198 |             →→→case 2:MOTOR_REVSTEP2;break;
199 |             →→→case 3:MOTOR_REVSTEP3;break;
200 |             →→→case 4:MOTOR_REVSTEP4;break;
201 |             →→→case 5:MOTOR_REVSTEP5;break;
202 |             →→→case 6:MOTOR_STOP;break;
203 |         }
204 |     }
205 |     →else //电机正转
206 |     {
207 |         →switch(step)
208 |         {
209 |             →→→case 0:MOTOR_STEP0;break;
210 |             →→→case 1:MOTOR_STEP1;break;
211 |             →→→case 2:MOTOR_STEP2;break;
212 |             →→→case 3:MOTOR_STEP3;break;
213 |             →→→case 4:MOTOR_STEP4;break;
214 |             →→→case 5:MOTOR_STEP5;break;
215 |             →→→case 6:MOTOR_STOP;break;
216 |         }
217 |     }
218 | }

```

图 3-8：换相函数

这里注意需要在换相时切换比较器中断边沿。

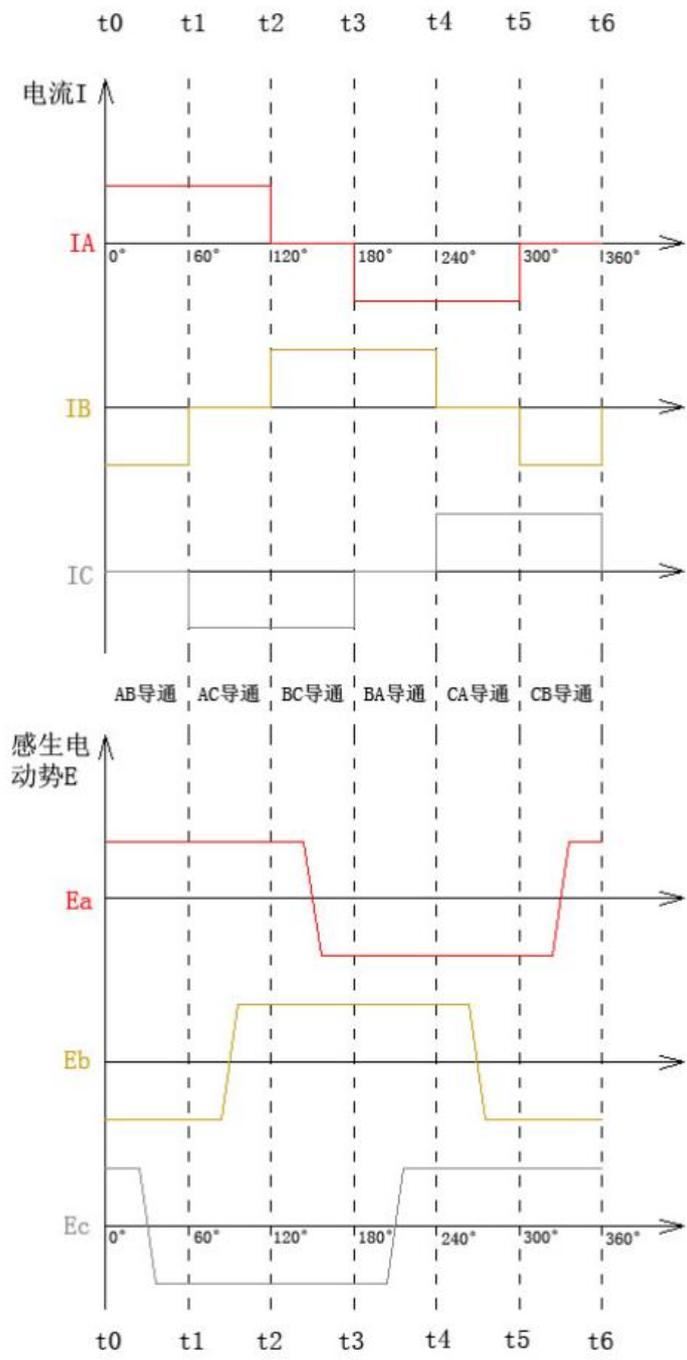


图 3-9：不同通电情况下各相波形和反电动势

```

69 //正转六步
70 #define MOTOR_STEP0 {APWM;BL ; ;CZ ; ;ADC_CONTR=0x83;CMPCR1_FALLING;}
71 #define MOTOR_STEP1 {APWM;BZ ; ;CL ; ;ADC_CONTR=0x84;CMPCR1_RISING ;}
72 #define MOTOR_STEP2 {AZ ; ;BPWM;CL ; ;ADC_CONTR=0x82;CMPCR1_FALLING;}
73 #define MOTOR_STEP3 {AL ; ;BPWM;CZ ; ;ADC_CONTR=0x83;CMPCR1_RISING ;}
74 #define MOTOR_STEP4 {AL ; ;BZ ; ;CPWM;ADC_CONTR=0x84;CMPCR1_FALLING;}
75 #define MOTOR_STEP5 {AZ ; ;BL ; ;CPWM;ADC_CONTR=0x82;CMPCR1_RISING ;}
76
77 //反转,中断沿极性取反,步序号反转
78 #define MOTOR_REVSTEP5 {APWM;BL ; ;CZ ; ;ADC_CONTR=0x83;CMPCR1_RISING ;}
79 #define MOTOR_REVSTEP4 {APWM;BZ ; ;CL ; ;ADC_CONTR=0x84;CMPCR1_FALLING;}
80 #define MOTOR_REVSTEP3 {AZ ; ;BPWM;CL ; ;ADC_CONTR=0x82;CMPCR1_RISING ;}
81 #define MOTOR_REVSTEP2 {AL ; ;BPWM;CZ ; ;ADC_CONTR=0x83;CMPCR1_FALLING;}
82 #define MOTOR_REVSTEP1 {AL ; ;BZ ; ;CPWM;ADC_CONTR=0x84;CMPCR1_RISING ;}
83 #define MOTOR_REVSTEP0 {AZ ; ;BL ; ;CPWM;ADC_CONTR=0x82;CMPCR1_FALLING;}
84 //停止
85 #define MOTOR_STOP → {AZ ; ;BZ ; ;CZ ; ;PWMZ;}
86 #define MOTOR_BRAKE → {AL ; ;BL ; ;CL ; ;PWMZ;}

```

图 3-10：换相逻辑

在启动过程中，由于没有传感器，无刷电机无法获得转子初始位置信息，通过反电动势获得换相位置也要求电机有一定的最低转速以保证采样的信噪比，因此采用三段式启动，即预定位，定压变频强拖启动，切入无感闭环；首先关闭比较器中断，任意两相通较低 PWM，然后强制周期性换相，并逐渐增加换相频率，以将电机强拖到足够快的转速，反电动势波形正常，再打开比较器中断切入闭环过零检测换相。

```

342 →Tim0_count2++;
343 →if(Tim0_count2 >= StartPeriod / 50) //启动时逐渐减小周期
344 →{
345     Tim0_count2 = 0;
346     →if((MotorState&FLAG_START) == 0)
347     →{
348         →if(RealMotorPwm >= STARTPWMMIN) //
349         →{
350             MotorStepRunNext();
351             StartPeriod -= StartPeriod/20; //逐渐增加强拖电频率
352             →→→if(StartPeriod <= 500) →→→ //当电频率大于200Hz时认为启动
353             →{
354                 MotorState |= FLAG_START; //置标志位
355                 MotorZeroCount=10;
356                 StartPeriod = 2500; //复位启动周期25ms
357             } →→→→→
358         }
359     }
360
361     →if(MotorZeroCount&FLAG_START)
362     →{
363         →if(MotorZeroCount<3) //若转速过低，则认为是停机
364         →{
365             MotorState &= ~FLAG_START;
366             MOTOR_STOP; //停机
367         }
368     }
369     →MotorZeroCount=0; //过零计数清0
370     →}

```

图 3-11：启动函数

在调节转速的过程中，注意使用的栅极驱动芯片耐压较低，因此需要限制电机转速变化率，防止泵升电压过高冲坏栅极驱动。

## 致谢

从 17 届开始构思这辆车到付诸实践，经历了很多很多波折，过年时候还在画板子，写无刷程序，那会儿换相逻辑有问题，在家里炸得满屋子烟雾，还好最后修好了，有了稳定的版本(虽然省赛前炸了)。

后来挨个零件设计加工，无数个日夜守在机床边上盯着，甚至中途机床 C 轴转台伺服电机烧了，返修要拆卸整个谐波减速寄回去修，于是干脆自己修，好几天才修好...这期间遇到了太多问题，在此感谢实验室的各个新老队员，每天忍受着加工的噪音(问就是把机床放实验室了，整个一层楼都是声音)；感谢各位老师提供的竞赛平台；感谢我的两个软件队友，他们虽然大一刚接触智能车，但是却付出了数倍的努力，成长得很快，校赛时也在其他所有做电磁的大二大三同学里面脱颖而出；感谢和我一起搞无刷的 17 届国一信标紫骅兄弟；感谢电磁王郭哥，智能视觉划时代凤凰结构的设计者凤凰哥-Phoenix，做硅胶轮子的 Ca，南风向北，帅帅，皮卡丘，华理轻工霍哥周哥，天科大龚浩然；感谢数个凌晨和我一起互相投屏画板的轩哥，很多脏活儿累活儿一个人做的时候是很孤独的，有一个人一起搞就更容易熬下去了。

最后的最后，感谢我的女朋友 YJR，是你看到了我所热爱的事业，和我追求的一切意义。无数个问题连环出现，想要放弃的时刻，想到你便有了坚持下去的勇气。感谢相遇。

尽管在自制的路上踩了无数坑，熬了数以百计的夜，按收益来算，可能不如使用现有车模去改装，但我还是会会在自己造车的路上继续走下去，而且不仅仅限于智能车竞赛。能够独立从原材料制造出成品，是一种终极的安全感的来源。

Handyman is never alone.

天津大学

马阔

2023 年 8 月