

清华大学第二十届电子设计大赛队伍报告

A_star

陈华玉 梁文浩 吾尔开希·阿布都克里木 吴紫屹

整体策略概述

小车主要由车身硬件系统，控制系统，策略系统，仿真系统与监控调试系统组成。

车身硬件系统包括整车设计，电源等外设管理（LED，键盘，编码电机，车轮，车身作图打印组装），引脚分布维护等。主要由梁文浩负责。

控制系统与策略系统分别由两块 stm32rct6 芯片协调完成（我们称之为控制板与策略板），板间依靠串口协议通信协调。

控制板主要负责运动控制，与传感器驱动配置与信息处理。依靠多级的反馈系统对整体硬件输出进行 pid 调节，提供多种工作模式（遥控控制，通信控制）与多种 debug 监视方式（用电脑端与手机端监视 debug）。控制板执行策略板的指令，保证将硬件（电机）输出能力发挥到极致，尽可能提高速度与精确度。控制板设计维护由陈华玉负责。

策略板分为架构搭建与导航部分。硬件架构主要是通信部分，包括与控制板通信，上位机通信与和 debug 工具通信，其主要功能是保证策略部分能够及时得到传感器信息并且把导航结果传递出去。此外架构负责一些规则方面的调整，如开场导引入道，急停，绕开敌人等等指令与导航协调配合，架构搭建部分由吾尔开希完成。

导航部分对地图进行拓扑化与参数化，并且在多人对战中快速检索最短路径，给出最优解，导航部分保证了策略与运动的宏观最优，保证运动板所做的运动是有意义的，导航部分由吴紫屹完成。

仿真部分，更改了上位机的源代码，通过手机热点模拟上位机，搭建了图形化仿真平台，这使得在没有控制板与场地的情况下，策略板可以独立调试 debug，在后来的实践中，仿真平台一次又一次证明了它存在的价值。仿真平台搭建由吴紫屹完成。

Debug 系统：debug 系统包括手机端 debug 与电脑端，手机端 debug 主要用于 pid 调试，在已有安卓蓝牙助手 app 上进行二测开发，将 pid 与其他可以自定义的数据波形显示出来，便于开发。控制利用手机可以直接遥控对小车进行控制，从而进行硬件检测。手机端 debug 开发有陈华玉完成、

电脑端 debug 利用已有程序二次开发，同样为通信助手，并且串口部分为仿真系统所用，由吾尔开希完成。

此外，吾尔开希由于处于多个系统协调配合的交叉点，广泛参与到了各个系统的设计调试与维护工作中，促进了多板多系统的协调配合职能。

-----陈华玉

控制策略概述 陈华玉

控制部分为一个 4 级级联的 pid 系统。利用陀螺仪信号，坐标信号，导航信号，编码器反馈与红外信号为输入。四个电机的 pwm 值为输出。利用蓝牙，led，按键进行模式选择与外界通信与 debug。

反馈流程：

1. 对电机死区进行校正，形成接近线性输出系统。
2. 利用码盘信号对每个电机转速进行 pid 调节，形成第一级反馈。
3. 对 mpu6050 陀螺仪信号进行 200hz 采样，而后低通滤波，而后交付 dmp 姿态解算（原理为开尔曼滤波），得到 yaw 值（后来更换了传感器为 jy61，但保留了 6050 接口）。
4. 利用导航信号坐标计算出目标点的距离与转角。
5. 利用坐标信号与目标距离，pid 设置车速，形成第二级反馈。
6. 利用 yaw 信号与目标转角信号对车速进行 pid 调节，形成三级反馈。
7. 利用红外信号校正车速，考虑多种情况，形成四级反馈。（由于红外硬件最后并不稳定，且地形复杂，我们最后并没有启用红外校正）
8. 其中第一级反馈为底层反馈，二三级反馈并行，第四级反馈在最顶层，只进行优化性质调节。
9. 总得来讲控制板核心为四级级联的反馈系统，其配备 debug 模式，调参模式，不同速度模式选择，监视信号输出，通信接收等多个接口，具有良好的可移植性与可拓展性。并且控制板对传感器与执行机构的信息或输出信号进行了平滑融合滤波等处理，保证了系统鲁棒性。一些具体的细节实现请参照代码。

Debug：

RgbLED 灯有多种显示模式从而指示车身状态，每一次烧录可以选择两种速度模式（决赛时我们采用的是快速与超快速组合）（利用按键），同时可以利用遥控控制，有与电脑手机和策略板通信接口（通信协议吾尔开希设计）。

硬件系统概述 梁文浩

1. 车体硬件(最终):

大功率编码电机 x4, stm32rct6 x2, L298N 电机驱动模块 x2, LM2596S 可调降压模块 x1, 小轮胎 x4, mpu6050 陀螺仪 x1, JY61 陀螺仪 x1, 摇臂开关 x1, 3S2200mah 锂电池 x1, ESP8266 WIFI 模块 x1, BC417 蓝芽模块 x1, 压克力板 x4。

2. 整体设计:

在前期我们做过几次试验,发现主办方提供的样车速度不高,没有太大的提升空间,原因在于电机的扭矩和功率都达不到我们的要求,于是我们商议改用大功率电机,并做成四驱的形式,大大提升小车的极限速度和加速度。

小车采用三层结构,是以功率、功能并考虑重心问题划分的。底层是大功率的动力系统,电机及其控制模块、降压模块等都会放在底层,几乎把所有重量都集中这里,使小车重心可以尽可能低;二层则是放置控制板和传感器等组件,把需要相互通讯的组件集中在一齐可以方便联机;三层都是一个用来放置颜色纸的平面。

3. 关于压克力板:

我们组的压克力板有两个来源,一是和编码电机配套购买,可以直接安装上电机;二是利用 AutoCAT 自己设计并切割。自己设计的板由于测量精度存在偏差和实际使用发现问题,中途也修改过好几次,也有由于战略原因放弃使用的配件。

4. 关于电路

小车电路其实并不复杂。动力系统方面是编码电机连接到电机驱动模块,然后直接连接到电源。信号系统方面则是电源接到降压把电压稳定在 4.9V 左右,然后才接到控制板和各传感器上。考虑到安全性和方便性,在电源输出口加一个大电流开关,以随时控制小车开关。

由于我们使用的是大功率电机,为保证电机转动时电池还有余力为其它组件提供稳定的电力,我们使用了大功率锂电池,而且由于电线也有自己最大电流限制,所以我们特地买了一粗一幼两种电线,考虑该线路通过电流的大小才选择使用不同电线,而只有在直接连到组件上的线才会使用杜邦线,以保证供电稳定。

5. 关于控制板接口扩展

虽然主办方提供了小型面包板让我们进行正负或其它接口的扩充,但面包板有占用体积大,所有线路都已固定,和排针插到面包板不牢固等缺点,所以为保证线路稳定性和空间利用合理性,我们只使用了一块面包板,其它扩充板都是用洞洞板自己焊接并切割成合理的形状的。

6. 关于维护问题

硬件的小问题很有可能在调试时会随机性出现奇怪的情况,为了使负责软件的同学能专注修改自己程序中存在的 bug,硬件必须要做到万无一失。

由于杜邦线质量参差不齐的问题,很有可能会导致线路连接不稳定,还有人手焊接也会导致焊点寿命和牢固性受影响,故需要定期检查并想办法使各组件连接稳定,为此在备赛期间曾三度大规模更换和整理线路连接。

导航部分概述 (吴紫屹)

由于此次比赛的地图较为复杂,且交通规则严格,稍有不慎就极易犯规扣分,因此在与负责小车控制的队友商讨后,我决定采用给点的方式来指示小车下一步要去的位置,而不是基于红外循迹之类(此块内容留待负责控制的队友阐述)。

考虑到 stm32 板的内存问题,不可能将地图直接二值化然后存入决策系统,且这样做也不利于跨区域决策,因此我的做法是将地图抽象为一些关键点(Position)、道路(Edge)、路段(Section)、路口(InterSection)的组合,通过有限状态机的思想进行区域划分,当需要寻路时,只需找到离起点和终点最近的地理单元,即可通过预计算好的路线(Path)进行路径规划,值得一提的是,路线(Path)结构体的组成为关键点数组,也就是一条折线段。

具体而言,我将地图分成了 40 个关键点,大多为弯道的起终点、路口的交界点等,在主办方给出的地图 CAD 文件中测出其坐标,存入 stm32 板子;然后用两个一组的关键点组成 26 条道路,也即将道路抽象为起点、终点和同样在 CAD 文件中测出的路长;接着用一系列相连道路组成 12 个路段,并记下此路段的相邻两路口,这是更高一级的封装;与此平行的还有 4 个路口,用于预计算一些路径。以上这些地理单元(除关键点外),都有根据坐标判断一点是否位于其中的函数,也是通过 CAD 文件测出来的。在完成地图划分之后,我预规划了 4 个路口之间的最短路径,同样存入 stm32 板。

有关实际比赛中的决策,我有两个核心函数 get_nearest_path 和 Search_Passenger。第一个函数传入起点和终点,规划出两点之间的最短路径,具体实施方法为,首先定位两点,找到其所在/最近且遵守交通规则的路口,然后以预规划好的路口间路径为主体,在前后加

上起点、终点到相应路口需经过的关键点，即成一条最短路径。第二个函数则是遍历场上所有乘客，调用 `get_nearest_path` 算出车当前位置到乘客起点的距离和乘客起点到终点距离，然后用此乘客的分值除以两端路径的总长，得到一个平均距离分数，以此作为选择乘客的标准。在具体实施中针对虚线掉头、躲避圆盘还作出了诸多细致的调整，此处不再一一赘述。

另外上述提到的仿真平台是基于开发组给出的上位机程序，添加 `SerialPort` 对小车传回的信息进行实时解码，直接显示在地图上。而与原版上位机地图外接摄像头不同，此处的地图就是用主办方提供的图片，解码显示的内容包括车判断的当前点坐标、下一个目标点坐标、二者之间的距离和夹角（与+x 轴方向）。此仿真平台对于策略部分的 debug 起到了极大的帮助。

主逻辑框架及上位机概述（吾尔开西）

1、上位机信息解码与维护

在串口中断函数中接收上位机传来的信息，每字节信息调用一次中断。用队列接收信息，利用信息头和信息长度来判断是否接收到完整信息，接收到一条完整信息后，将队列里的信息复制到数组中，对数组中的信息进行解码，并判断比赛是否暂停或结束，车是否应该停下。解码是在定义的一些全局变量和结构体中存储上位机信息。

出现的问题与解决方法：

（1）开发初期，没有注意中断优先级的问题，在有多个中断函数时，不设置中断优先级将会使掉包率大幅上升，导致小车得到的有效信息数不足，无法正常控制。发现后将接受中断的优先级提高，问题解决。

（2）由于在中断函数中对上位机信息进行解码时会更改一些全局变量，而在主逻辑中可能会用到这些全局变量，若在使用时全局变量的值被更改，则会导致一些问题。所以设一个标志符，用来控制是否在接收中断解码，问题解决。

2、两板通信

由于开发初期采用 `mpu6050` 陀螺仪，单片机被占用的资源较多，所以决定用两个单片机分管决策和控制。两板之间用串口通信，决策板将小车当前坐标，目标点坐标、一些控制信息和校验位存在一条长度 10 字节的数组中发送给控制板。

出现的问题与解决方法：

（1）控制板接收中断优先级设置错误，导致掉包率过高，控制板不能有效控制，改变优先级后问题解决。

（2）决策板编码时没有采用有效的位操作，导致在一些情况下编码错误，且错误不易被察觉，改变编码方式后问题解决。

3、逻辑主循环

比赛开始时，先通过特定引脚的高低电平判断自己车的身份（A 或 B 车），并给出相应的当前坐标和目标坐标，使小车从仓库走到道路上。进入主循环，先判断我方车上有没有乘客，若有乘客，则计算出到乘客终点的路线；若我方车上没有乘客，则选择最划算的乘客，并计算出路线，并依次将路线上点的坐标发送给控制板，途中，若乘客信息或比赛信息发生变化，则回到主循环开头，同时避让对手。当前路线上的点全部走完后，回到主循环开头。

陀螺仪

后期，mpu6050 发生严重偏移，改用 jy61。Jy61 有自己的计算芯片，只需要通过串口读入角度信息即可，接收中断的设计与上位机信息接收类似。

4、避让对手和圆盘

赛场左下角有一个圆盘，小车不能踩在圆盘上。由于路线计算过程较复杂，所以计算时没有考虑避让圆盘。在路线计算出来后，检查路线上是否有连续两个点连成的直线跨在了圆盘上，如果有，则用向量计算的方法计算出一个点，插入到路线中。

主循环进行过程中同时要考虑避让对手，涉及的各种情况较复杂。

- (1) 若我和对手都在同一条单行道，且对手在我前方，挡在了我和目标点之间，则停下。
- (2) 若我在单行道内，而对手在单行道路口处挡住我，则停下。
- (3) 若我和对手都在可自由行驶的十字路口内，则将对手看成有一定半径的圆盘，计算出一个临时点用来绕过对手，判断该临时点是否出界，如果没有出界，则前往该临时点，如果该临时点出界，则计算另一边的临时点，进行同样的判断，如果两个临时点都出界，则停下。

