2015年飞思卡尔智能车竞赛（厦大）总结

2016年1月2日

本次比赛，我们使用的是E车模。从硬件设计，到电路板的制作。到单片机的初始化和，整个程序运行流程的设计和实现，到最后的pid参数的整定和特殊情况的处理，完全是由我们全组成员一起共同完成的。由于有着去年的经验，很多问题都可以提前想到，使得我们的进度也比较快，少走了很多弯路。但是也存在很多不足之处，这里我将我们组的设计思路和赛后的一些想法和建议和大家分享，希望对大家有所帮助。

首先硬件上，从学习画pads到布局布线，到最后投板生产，是一个考验细心和耐性的过程。一个人画是很容易出问题的，很多问题欠考虑，所以建议两个人一起画，画完多找人看看，不然投板生产，焊接完成后出现很多问题的话，都得重头开始，这是一件让人绝望的事。还好我们没有遇到这么倒霉的事情。建议所有人都要参与进来，软件的接口设置，硬件的布局也会影响机械结构。所以一定要统筹考虑，胆大还要心细。多积累资料，从前辈的板子上学习，借鉴。

程序上我们整个工程主要是cintrol.c和anglecontrol.c两个主要文档。主要用了一个1ms的中断。在主函数main()中执行20ms的循环。我们使用了两个CCD一个采集远处的信息（60CM），一个采集近处（40CM）近的负责转向，远处的则协助近处的做赛道类型的识别或者在近的ccd丢线时使用其位置信息:

直角：当近的CCD看到黑线，记录标志延时，延时内大前瞻一边丢线，并且小车位于赛道中央（避免弯道误判）则判断为直角。对于不同的速度给予不同的转角，并保持一段时间。

单线：从循线算法上下手（使用提取下跳变沿的方法）将从上次黑线位置处开始改为从上次黑线偏右一点向左寻找左边线，偏左向右一点开始寻找右边线。取中间位置。

障碍：判断障碍物与边线之间的较窄的白色（漏判率较高）需要多考虑；

十字：十字转弯时容易走错道，尝试过很多方案，但是都只能是有一定作用不能从根本是解决问题，最好的办法也是通用的方法还是调整或者使用更加优越的pid参数或方法。我们的方法是：当十字处有一边丢线容易当成普通弯道走错道，但是当小车再往前走就会看到两条道，那么这时，我们就可以选择靠近丢线方向的那条道。

方向算法中我们将偏差量使用了一段二次函数来实现非均匀变化，小偏差小变化，大偏差大变化。

direction\_Erro[0]=0.8\*(black\_line-66)+(black\_line-66)\*(black\_line-66)/100;

效果很好。

最后的心得，pid的整定非常重要，在路径识别上可以省很多事。对于直立车直立很重要，电源对pid性能的影响很大，而我们学校没有给电机电源稳压的传统，但是观察中南的车有电源稳压模块。根据我们调试的经验，电池电压对直立影响很大，建议对车模加上电机稳压模块，这样对小车pid的整定会有很大的好处。其他组别电池电压或多或少多会有影响，对于高速行驶的小车这样的误差不容小视！