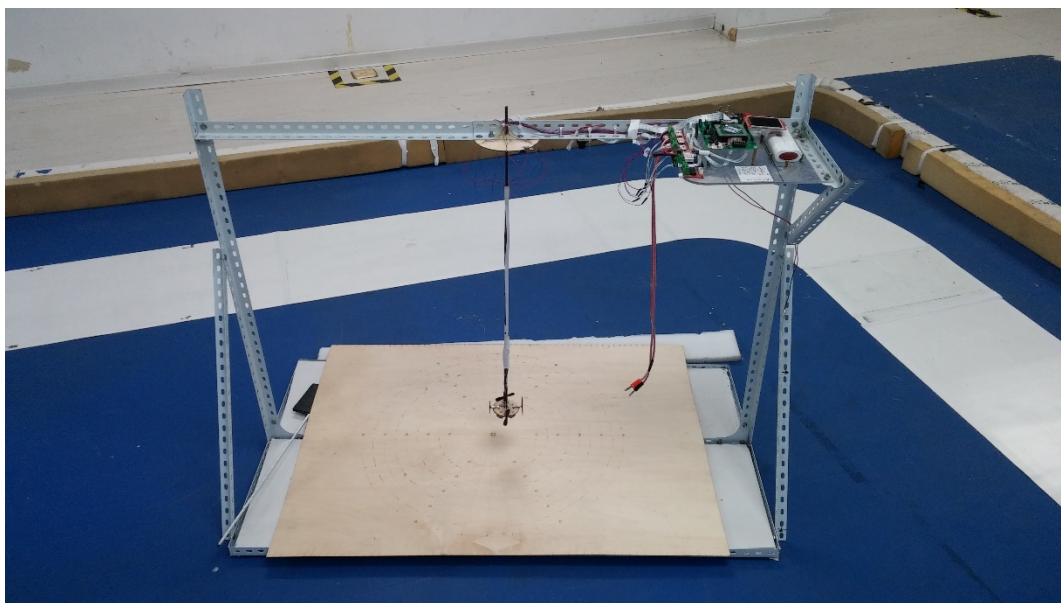
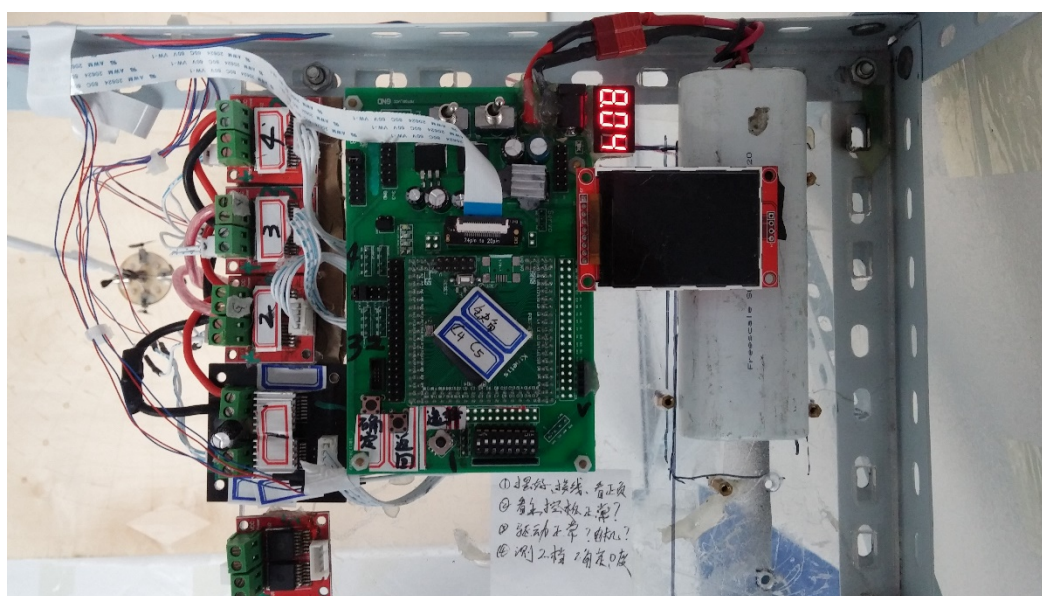


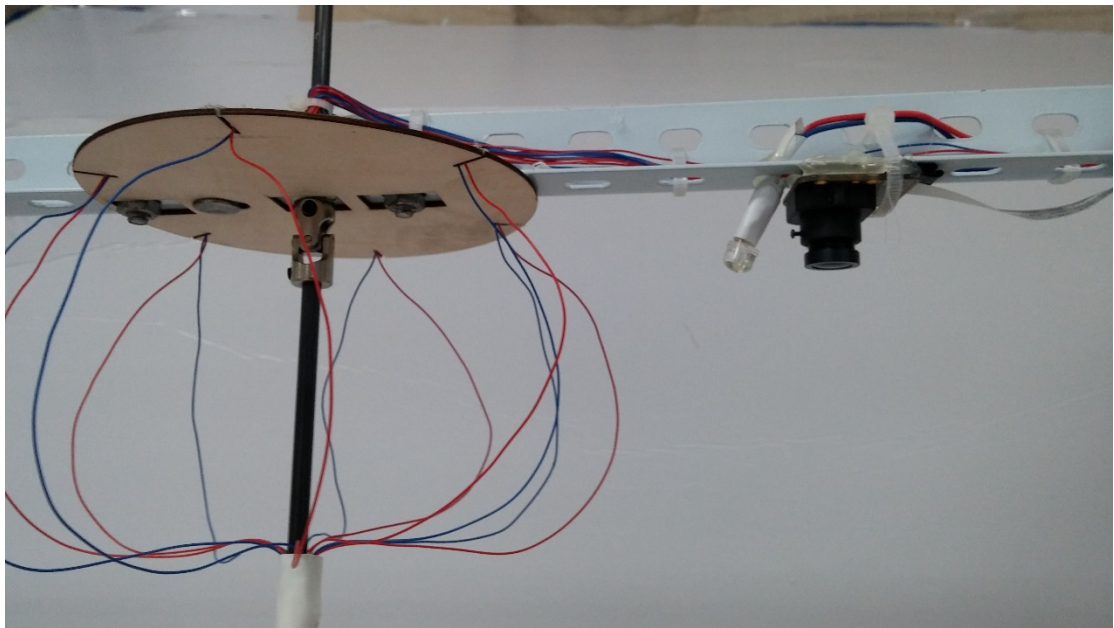
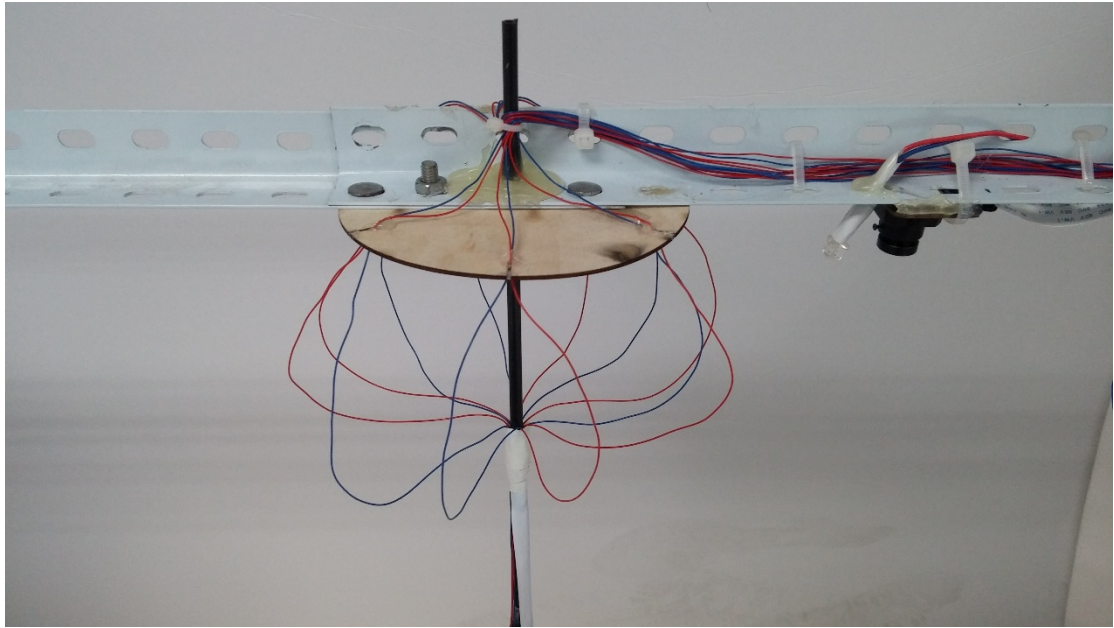
下图是作品的整体照片，使用的是一个展板的架子作为主架，坐标底盘用激光雕刻机雕刻，背景是智能车赛道。



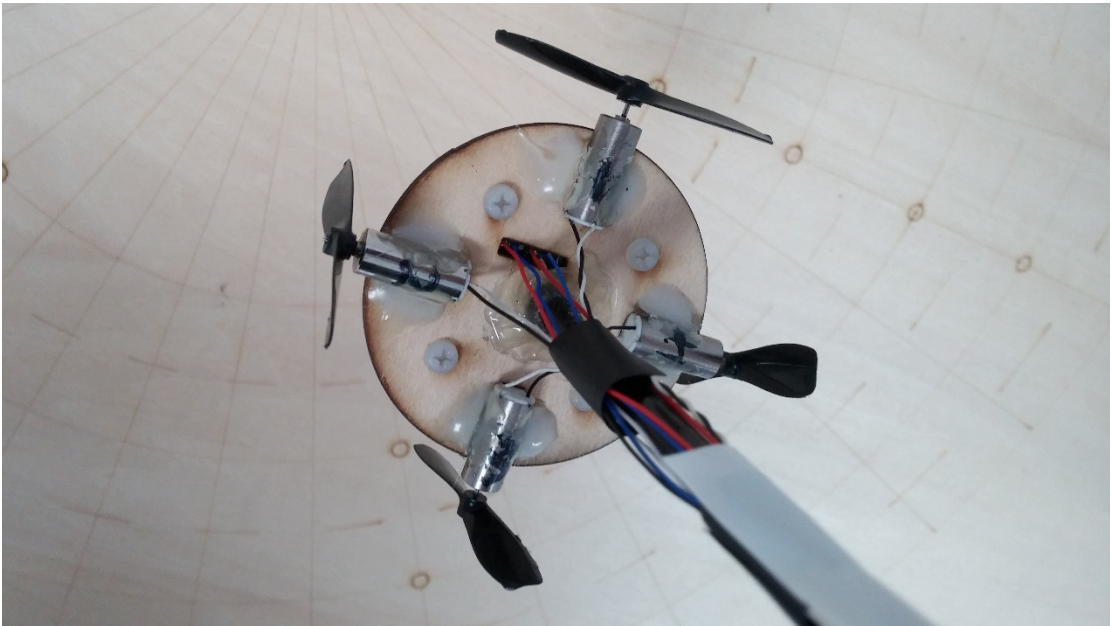
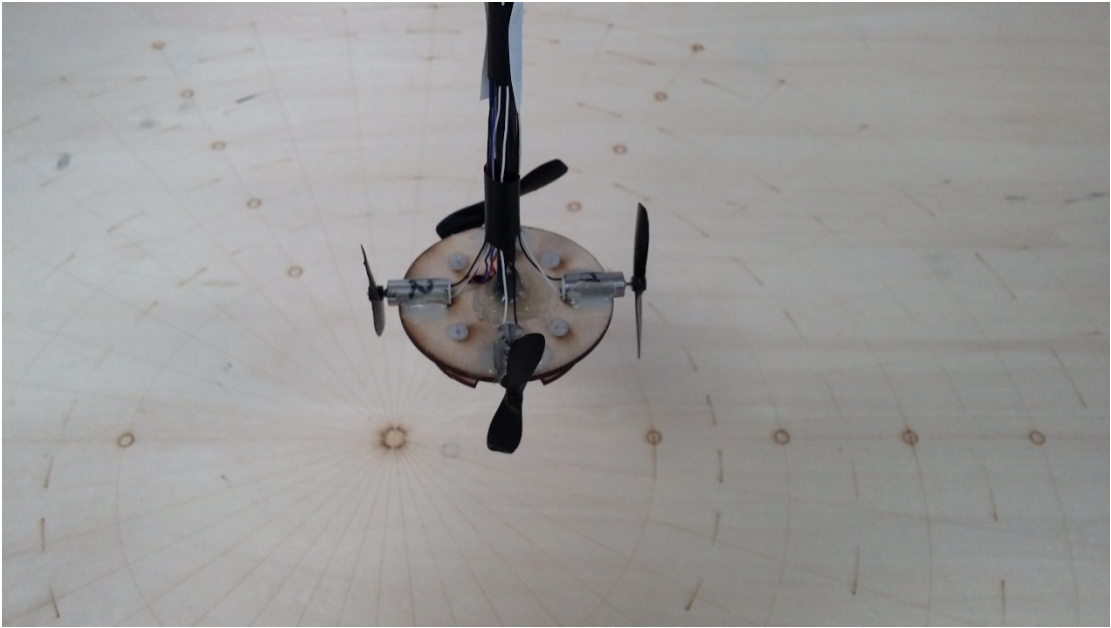
下面是硬件图，比较丑，但是功能全都有，左边 4 个驱动板，备用一个，拍摄于比赛还未结束，所以比较丑，封箱的时候比较整齐。

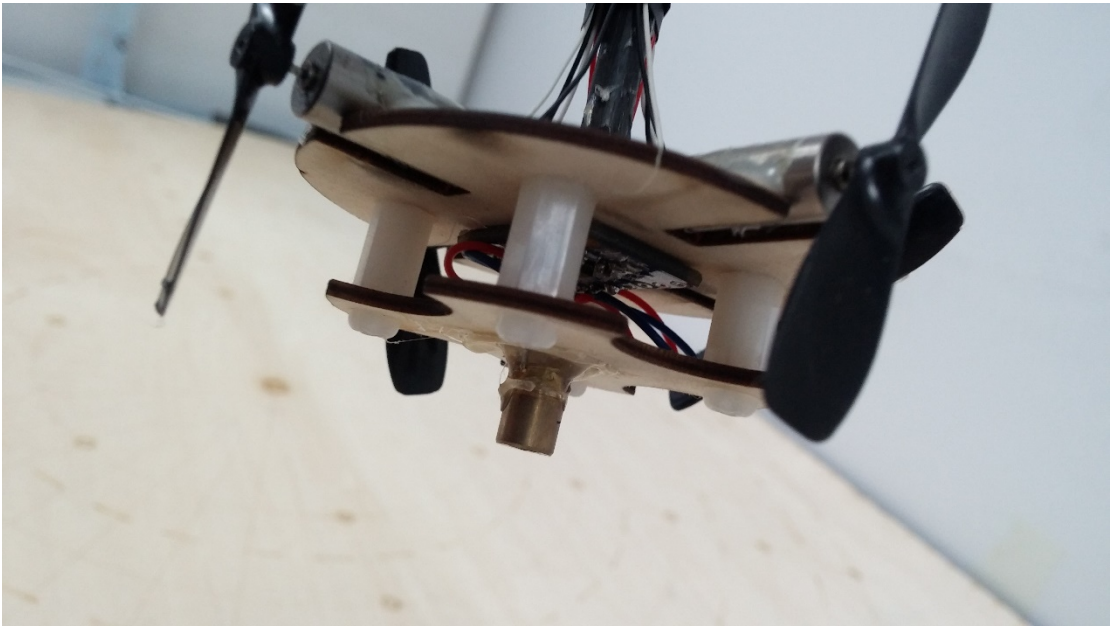
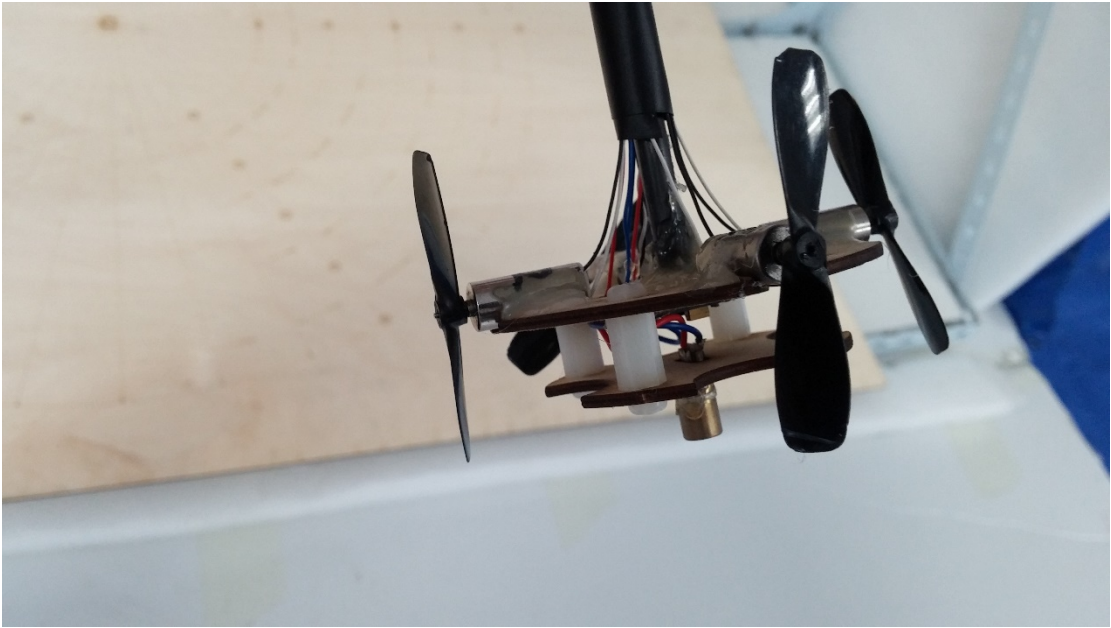


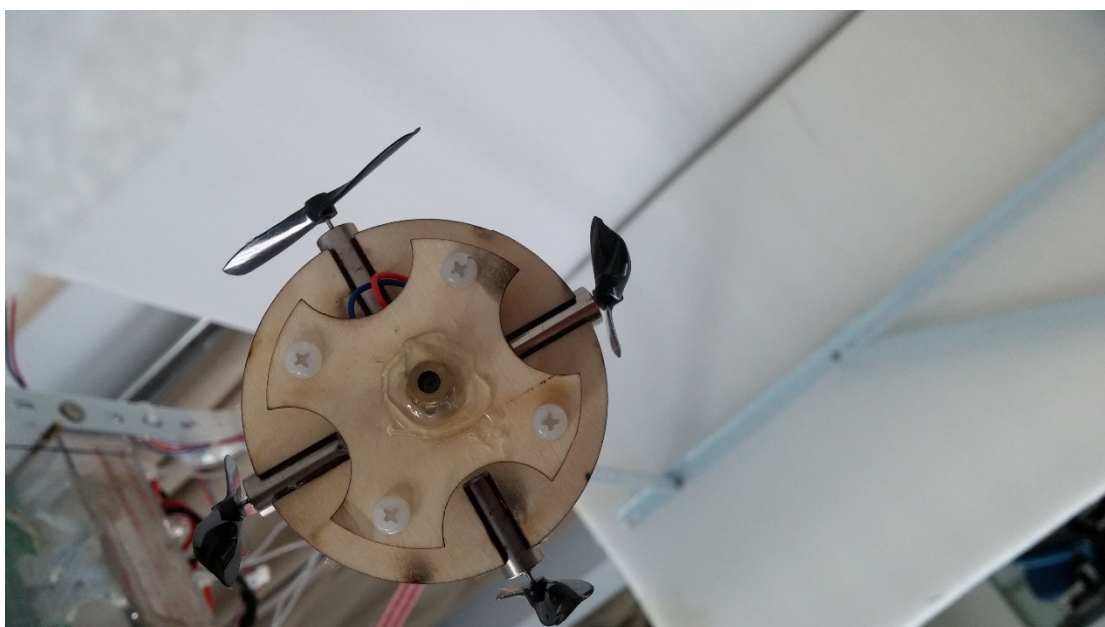
下面两个图片是上托盘上下表面，线要用细线，而且要很称对，否则会很干扰运行，看似线没有什么力，但是杆子受到了线很大的力，造成不对称。



下面几张是下部托盘的图片，两个激光雕刻托盘，一大一小，上面放电机，中间放 MPU6050，下面是激光。







做了这几年智能车和电赛的经验来看，我认为做智能车或者做控制题目，比例是这样的：2 成硬件+3 成机械+3 成软件+2 成传承。没算上重要的“人自身的主观能动性”，因为无法衡量。

1.硬件：

1. **单片机**：K60，优点不解释，缺点是有的引脚速度不一致，用的是软件 IIC，所以这个缺点有点坑爹，有的时候 SCL 和 SDA 调换一下就不行了。。想不通什么，反正归为速度不一致的 bug 了，小伙伴们注意啦。

2. **姿态传感器**：MPU6050。集成加速度传感器和陀螺仪，安装比较方便。买淘宝 10 块钱的那种就行，也不用买什么集成串口的那种，那种贵而且速度慢，就买个裸的模块用单片机读出数据自己做融合就行。

3. **其它**：除了前两个，感觉其它都不是很重要，用什么都行，只要稳定就好。我的观点是硬件最重要的是稳定，不管多贵或多便宜，一定是稳定第一。

4. **主控板硬件组成**：飞思卡尔电池、稳压 5V 模块、稳压 3.3V 模块、K60 核心板、LCD 显示屏、蜂鸣器电路、4 个 LED 灯、蓝牙接口、IIC 接口、2 个按键、1 个五向按键、8 位拨码、4 路 PWM 驱动、2 路正交测速、摄像头接口、ccd 接口、单片机所以引脚引出。

5. 还用到了驱动板电路 4 块，每块 2 个 BTS 芯片。

硬件不多说，只要能行就行，总之是稳定！稳定！！再稳定！！

2. 机械：

机械这个东西就很难说了，做不同的东西需要不同的机械。但是有这么几点是一样的：

1. **用心去做**。这点说起来就是废话嘛。。。但是，无数的经验教训告诉我们，一定要用心去做机械，即使是拧一个螺丝也好，否则会拿好几倍的时间去补偿的。我感觉去年我能拿直立国一的原因就是三个人做事都很认真，甚至是强迫症，在此感谢下亲队友，祝你们学业有成，勿念=。=。举个例子，马马虎虎做机械可能 2 个小时就对付完成了，但是调着调着就散架了，修了一个小时，又调了会，又坏了。最后坏到不能修了，不要以为我开玩笑，这种场景一直在做智能车和做电子设计中出现。我本人是写软件的，但是我对硬件和机械的要求非常高，就是要保证软件好调试，能有个好的硬件和机械调试是很舒服的。

这么长时间以来，我的总结就是：**机械搭不好，软件再厉害也弥补不过来**。

就算你是调 pid 的大神，面对屎一样的机械会无能为力的，反正带智能车的时候学弟找我调试，我看到硬件不稳定或者机械比较渣的话我就不会帮忙。其实可能只要把机械搭好一点问题就解决了，通常就是这么神奇。

2. **对称**：这点相当之重要啊，智能车要对称，风摆要对称，硬件要对称，什么的都要对称。这是调试软件的重要前提，经验总结就是：电机要对称，轮子要对称，摩擦力要对称，重心要在中央，镜头要对称，陀螺仪要对称，等等等。

举个例子。调过直立的小伙伴都知道陀螺仪歪了会有什么致命的效果：左拐和右拐会有直立方向的分量，让小车以为自己在上坡和下坡，也就是一个加速一个减速。此时掰一掰陀螺仪就好了。

3. 这次的风摆也是，4 个电机一定要对称，因为不管是直线、斜线、圆，基础都是横着摆和竖着摆要摆好，不对称的话会造成夹角不是 90° ，会对斜线和圆造成很大影响。**所以要 4 个电机对称，保证电机型号一样，斜对的电机在一条直线上，两条直线一定要垂直**。我们学校用的激光雕刻制作的底盘，做好了 4 个

凹槽，所以安装简单些，对称些，不过还是要微调好久才比较对称，所以不要着急，一定要每个步骤都认真完成。还有拿 pcb 做的托盘，效果也不错。

3. 软件

准备工作：测量沿着 x 轴摆动的偏角，就用《官方直立车参考方案 2.0》上的方案就行，加速度计数据和角速度融合。y 轴同理。其实不用想的高大上，什么姿态解算啊，四元素啊，其实只是直立车的角度测量代码执行两遍，测两个方向的角而已。**参数：**1ms 进一次中断，奇数 ms 软件 IIC 采集两个方向数据并滤波，偶数 ms 角度融合、PID 计算、电机 PWM 输出，所以是 2ms 一个控制周期周期。以下是定时器中断代码。采集数据 MPU6050_updateSensor 和数据滤波 Filter_2nd_LPF2ndFilter 的代码和直立车代码一模一样，只是用的是 MPU6050，所以不贴出来了，获得两个方向的角度 Angle_Calculate() 和 pid 计算 Mode_PID() 和电机输出 Motor_Out() 请看附件 ISF_fun.c。

```
void PIT0_IRQHandler(void)
{
    PIT_Flag_Clear(PIT0);
    Time_1ms++;
    Water_LEDs(); //LED 流水灯，证明在进定时器中断
    //姿态解算
    if(Time_1ms%2==0)
    {
        MPU6050_updateSensor(); //600us//采集数据
        Filter_2nd_LPF2ndFilter(); //20us//数据滤波
    }
    else
    {
        Angle_Calculate(); //获得两个方向的角度
        Mode_PID(); //pid 计算
        Motor_Out(); //电机输出
    }
    PIT_Flag_Clear(PIT0); //清中断标志位
}
```

方案备选 1：强行摆动。就是用 pid 控制位置，就是指哪打哪里，基本上就是列一个 (x, y) 表，以横摆为例，设置为 (0, 0) (1, 0) (2, 0) (1, 0) (0, 0) (-1, 0) (-2, 0) (-1, 0) (0, 0)，这样重复下去。我这里只是随便说说，数据表应该换成非线性表或者直接用公式什么的我也不太清楚，思路大体是这个，但是我们学校用的不是这个方案，但是隔壁海事是这个方案，这种方案摆动周期比较长。但是简单粗暴，隔壁海事一天半就做完了，两天内就封箱了。没错，我说的就是海事小哥，智能车和电赛双国一。十分佩服技术和人品，技术过硬，也善于交流，对我们学校帮助也很多，祝其补考通过，保研北航顺利吧，还是单身哦，抓紧找对象吧，

研究生就不好找了，我看离你家很近的那个太原理工的直立妹子就不错=。=。说多了，那就再多说一句吧：大家做东西要多交流，共同进步是极好的，共同促进技术的提升。再次由衷感谢海事小哥，真的帮了很多很多。来长沙包吃住玩哦~希望你不要看到这段。。。

方案备选 2: 按照物理规律摆动，大家都学过单摆，B 题的机械搭出来，自然摆动的周期基本上都是 $1\sim 2\text{s}$ ，而且每次摆动只是损失很少能量，所以只要每个周期电机吹一点点就行了，把能量补充上就好了嘛。周期设为了 $T=1500\text{ms}$ 。

具体操作请看以下步骤：

基础部分第 0 问：大家要问为什么有个 0 问呢，不是基础部分只有 1 到 4 问吗？而且这段字体要标红，加粗和斜体。这是因为这个第 0 问才是最最重要的，比其它哪问都重要。**这问的目标就是：**在 x 轴上，随便指定某个角度（比如说 $+10^\circ$ 或者 -5° ），然后定在那里不动。**目的是调好 PID，因为以后所有问都不用改 PID 了。**

步骤：测量 x 轴的角度（ $-90^\circ < \theta < 90^\circ$ ），使用 PID 算法计算电机的 pwm 输出，控制 x 轴方向的两个电机，一个推，一个吸（就是往一个方向用力）。

注意 1：加速度计和陀螺仪的零偏一定要减去，解决方法就是：上电前扶稳传感器，一动不动后上电，采集数据很多遍，然后做个平均得到零偏，以后加速度和陀螺仪的值减去这个上电采集的零偏值即可。

注意 2：加速度计数据滤波要做好，大家都知道加速度计毛刺比较多，多次取平均等低通滤波方法均可，然后再和上次的值做个平均，这相当于再次滤波，这个平均值保存为当前值，就可以用了。

注意 3：角度融合，使用《官方直立车参考方案 2.0》角度融合即可，使用的代码就是这个里面的代码。

注意 4：PID 调好，调这个靠的是经验，所以还是要多做比赛实际体会怎么调试才行。用的是位置式，I 的积分值加了限幅。PID 调好要有以下两个现象：1、**稳定**，稳定在指定角度不会随着时间变化而跑偏，用手使劲波动，在一定时间内能回到原处。2、**跟随性好**，比如说处在 10° ，然后设成 -5° ，看看能不能快速地过去。好的 PID 参数至少要满足以上两点才行，如果不满足以上两点，PID 是不合格的。

PID 参数如下 {100, 3, 52}。PWM 满量程是 ± 4000 ，6V 电压供给驱动模块，4 个电机限幅均为 ± 2400 。P 乘以目标角度和真实角度的偏差，单位是度，I 乘以积分累计值，限幅在 ± 2800 ，D 乘以角速度，单位是度/秒。

同样的方法也把 y 轴调好，如果电机一样，搭的对称的话，用同一套 pid 即可。如果 x 轴和 y 轴不是垂直，请修改机械，直到对称为止。

基础部分第 1 问：如果 PID 调好了，而且比较对称，就可以调第一问了。公式：

```
Angle.GoalL = Angle.ForceL[Angle.Length] * sin((Time_lms - Mode.TimeStart)*2*Pi/((float)(Angle.Period)));
```

```
Angle.GoalH=0;
```

来解释下上面这个公式：

Angle.GoalL: 其中 L 是列 (Lie) 的意思, Angle.GoalH 中 H 是行 (Hang) 的意思, 就是前面总提到的 10° 、 -5° 这些目标角度。

Angle.Length: 数组下标, 单位是 cm, 代表要设置的摆动的直径, 这一问设成了 60, 因为题目要求不少于 50cm。

Angle.ForceL[]: 是一个数组, 表示不同摆动长度下的振幅大小。

Time_1ms: 是一个软件计数器, 每 1ms 自加 1。

Mode.TimeStart: 因为有一个开始的按键, 这个按键按下时 Mode.TimeStart 就记录此时的 Time_1ms。

Angle.Period: 1500, 单位 ms, 应该直接就能用, 效果不好请实测周期, 就是自然摆动 n 次的时间除以 n。

看到这个公式你可能有疑问, 这个第二套方案不是和第一套方案基本上一样吗? 前面基础部分第 0 问是固定目标值, 这个第一问只是让目标值按照一个周期以 sin 变化而已嘛, 只不过是用公式算罢了, 和第一套方案没有什么区别啊。

看似是这样没错, 但是请想一下: 如果是第一问的话, T 会比较大, 观测效果是: 角度变化会很生硬, 就是一点一点变化过去的。假设想摆动长度 60cm 时对应的角度是 $\pm 15^\circ$, 第一套方案的 Angle.ForceL[60] 应该是 15, 接下来问题来了, 使用第二套方案的话, 实测效果是: Angle.ForceL[60] 应该取比 15 大的数, 可能是 20 或者 25。

具体原理我也没有想清楚, 毕竟我不是学物理的, 是学计算机的, 不过我猜想应该是惯性的问题, 就是第一套方案周期长, 运动的很慢, 基本上每个点速度都很小, 所以强行调 pid, 强行设置位置, 指哪打哪即可, 说白了就是走一段停一段, 所以看起来也不是很连续。**第二套方案是模拟自然摆!!!** 只是在每个周期提供很少的能量罢了, 按照无阻单摆计算的话, 遵循能量守恒, 最上边是速度为 0, 最下边速度最大。此时的模型不是一个强行指哪打哪模型, 而是一个比较复杂的, 为实际单摆系统持续提供能量, 使其模拟无能量损失的单摆模型, 能一直摆下去不停止。**有哪位大神按照我的思路做了出来的话可以和我交流下, 我确实是不知具体的受力分析是怎样的。**

以上原理分析只是我个人的猜测, 不管怎么样效果还是很好的。

基础部分第 2 问:

第一问是第二问的特殊情况, 将这两问一起做了。此问要求摆动长度可设置, 可以取 Angle.Length 从 20 到 70 变化, 然后改变 Angle.ForceL[Angle.Length] 这个振幅的值, 直到每个 Angle.Length 摆动大小合适为止。其实就是做一个 x 轴的对应表。长度 Angle.Length 和振幅 Angle.ForceL[Angle.Length] 的对应表。Angle.Length=60 就是第一问的答案啦。

然后同样的方法设置 Angle.ForceH, 做一个 y 轴的对应表。y 轴也要测是为了基础第 3 问和画圆部分。

基础部分第 3 问:

```
cosAngle=cos(0-  
(Angle.DirAngle*10.0+Angle.DirErr[Angle.DirAngle])*2*Pi/360.0);
```

```
sinAngle=sin(0-  
(Angle.DirAngle*10.0+Angle.DirErr[Angle.DirAngle])*2*Pi / 360.0);
```

```
Angle.GoalL = cosAngle * Angle.ForceL[Angle.Length] * sin((Time_lms -  
Mode.TimeStart)*2*Pi/((float)(Angle.Period)));
```

```
Angle.GoalH = sinAngle * Angle.ForceH[Angle.Length] * sin((Time_lms -  
Mode.TimeStart)*2*Pi/((float)(Angle.Period)));
```

Angle.DirAngle: 取 0-35 这 36 个整数，就是基础部分第 3 问要设置的方向，0 代表 0° ，3 代表 30° 。

Angle.DirErr[]: 是一个矫正数组，先忽略之。

看到这里应该可以很容易猜到，其实就是物理上合力的模型，用 x 轴和 y 轴的力去合成一个新的力，这个单摆题也是一样。Angle.DirAngle=0、18 时，就单纯是 L（列）方向的运动，Angle.DirAngle=9、27 时，就单纯是 H（行）方向的运动。

但是，当你设置 20° 时，可能是 18° ，此时需要一个矫正表，那就是 Angle.DirErr[]，此时 Angle.DirErr[2]=2。这个矫正表是按照某种规律变化的，因为合力的原理是没有错误，但是设置的目标角度值可不是力，具体物理原理也搞不清楚，但是肯定是存在一个公式推导过去，花了 2 个小时从能量守恒的角度进行推导，废了 n 张草纸，太麻烦了，也没推出来，本来想用公式替代这个矫正表，但是还是算了吧，不钻牛角尖了。。。原谅高数只考了 100 分。

还有一点就是 0° 和 180° 的矫正数字是一样的， 10° 和 190° 的是一样的。以下是偏差矫正表，不过不同机械的数值不一样，贴出来只是想让诸位大神做出来的话联系我，交流下是不是所有的矫正表的规律都一样：

```
Angle.DirErr[0] = 0;  
Angle.DirErr[1] = 1.6;  
Angle.DirErr[2] = 2.5;  
Angle.DirErr[3] = 2.5;  
Angle.DirErr[4] = 2.0;  
Angle.DirErr[5] = 1.0;  
Angle.DirErr[6] = 0.5;  
Angle.DirErr[7] = 0;  
Angle.DirErr[8] = -1;  
Angle.DirErr[9] = 1.0;  
Angle.DirErr[10] = 2.0;  
Angle.DirErr[11] = 2.0;  
Angle.DirErr[12] = 1.5;  
Angle.DirErr[13] = 0.3;  
Angle.DirErr[14] = 0;  
Angle.DirErr[15] = -1.5;  
Angle.DirErr[16] = -1.5;  
Angle.DirErr[17] = -2.0;
```


基础部分第 4 问：

这问是最好做的！！

以下是第四问代码：

```
else if(mode == 4) //第四问专门设置
{
    Mode.P = Mode.para[mode][0] = 100;
    Mode.I = Mode.para[mode][1] = 0;
    Mode.D = Mode.para[mode][2] = 200;
    Angle.ErrorI_H = Angle.ErrorI_L = 0;
}
Angle.GoalL = 0;
Angle.GoalH = 0;
```

第四问不能有积分值 ErrorI，I 参数设置为 0。

D 参数最重要！！！要很大很大，大到出现高频抖动再减小点，原理很简单嘛，这一问就是不让它动，只要动就会有角度和角速度，所以 P 和 D 要大。

P 参数也要大一点，更加快速达到静态点。

经实测，即使偏到 90° ，放手后均能在 2s 内静止不动在静态点。

发挥部分第 1 问：

大家应该学过李萨如图形吧，或者学过解析几何， $x^2+y^2=1$ 表示的就是解析几何的圆。

```
Angle.GoalL = Angle.ForceL[Angle.Length] * cos((Time_lms -
Mode.TimeStart)*2*Pi/((float)(Angle.Period)));
```

```
Angle.GoalH = Angle.ForceH[Angle.Length] * sin((Time_lms -
Mode.TimeStart)*2*Pi/((float)(Angle.Period)));
```

这两个公式其中一个和基础部分第二问的公式一样，其中一个方向按照 \sin 变化，另外一个方向按照 \cos 变化，让相位总是相差 90° ，这不就是个圆嘛，振幅是一样的。经实测，3s 内基本上圆形就摆好了，10s 内肯定就稳定啦。

发挥部分第 2 问：

没有做，直接拿发挥部分第一问上，看不出有任何影响。这套机械和软件貌似还是比较吊的。最开始纠结让不让用空心杯，但是买了快 100 个的大型风机根本就吹不起来啊！！！全是钱啊！！

发挥部分第 3 问：

感觉能拿下瑞萨杯还是主要靠这一问，这一问做的是一个智能跟随系统。。。听起来比较高大上，其实却蛮简单的。就是下面铺一个白色的背景，用一个头部是黑色的“苍蝇拍”放在底部，拿一个摄像头做识别，识别出黑色的“苍蝇拍”头，然后控制电机指向这个“苍蝇拍”，让激光照到它。效果就是随着苍蝇拍的运动，激光点能够跟随过去。启发来源是谷歌之前说过用无人机送披萨。

4：控制题-赛前准备

4.1 人员分工:硬件、机械、软件，这三要素至少其中有一个人过硬，不管这三点分布在哪个人身上。队长最好实力好些，最好有强迫症，要求硬件机械软件这三点都苛刻些，尤其是机械，最好做事果断，但是要听取队员的意见。组队不忌讳拌嘴打架，忌讳三人意见不统一，好的模式是先集体定好一个方案，然后按照各自优势进行分工，方案不可行再换嘛，不要每个人搞各自的方案。不管怎么样，要多参加比赛进行磨合才是王道。

4.2 硬件:

主控板:提前做好含有**所有功能的主控板**，引出电源接口(不要用一般的 1117)，核心板可拔，引出所有单片机引脚，参考 2015 高元龙电赛硬件。准备两块即可。单片机核心板多备几个。

驱动模块: BTS 即可。准备 4 个即可。

蓝牙模块: 至少一对。

传感器: 加速度计，陀螺仪，绝对编码器，相对编码器，霍尔测速传感器，CCD，摄像头，光电管，等等。越多越好。

显示屏: 越多越好。

电机:越多越好。

推荐一个人备着就好，大家用的时候再拿，不推荐每个队备自己队的东西，作为一个集体，集体整体好大家才会一起好。

4.3 软件:

调 PID: 要求是精通调 PID，位置式和增量式。

准备的硬件，对应的软件写好: 将准备的所有硬件均写好软件，即整合在一个工程里面，保证比赛时候能直接调用。比如说各种传感器的代码，测速，pwm 输出等等。

4.4 官方准备清单

这点是重中之重，因为看到清单就可以猜题了，而且好多传感器和器材临时采购来不及，所以不要怕浪费钱，一定要多买多买再多买。比如说 2015 的清单，有万向节，直流风机，还有风速测量仪，这种一猜就是控制题的东西一定要多买。虽然有可能用不上，但是用上了就值了，否则临时买真的来不及，比如说万向节，临时买根本买不到，只能网上买，所以多多益善。

有时做电子设计竞赛拼的不仅仅是实力，还有购买力。所以东西一定要准备好，谨记。

附件 ISR_fun.c 是定时器中断调用的几个函数。

《官方直立车参考方案 2.0》下载链接: <http://pan.baidu.com/s/1o6lmeXs> 密码: flpx

有机会传几个视频给大家看吧，时间匆忙没来得及拍。