



# 开发日志

队名：大连八中救援队

参赛单位名：大连八中

队员：郭琪 关正 戴子龙 邓少淳 李欣然 郭婧汝 于佳冬 彭骏涵

张开源 付郑语

提交日期：3.10

## 1 讨论过程

(说明：请详细记录每位小组成员的发言，并通过草图的形式，记录每位同学的意见；请附图片)

郭琪:今天我们进入了下一个阶段的研究了,在第一个阶段我们不仅针对规则和各个电子元件展开了讨论与研究,还根据规则设计了相关的器件。

戴子龙:是啊,但是上一个阶段的很多猜想目前还处于试验阶段,并没有成形。

关正:而在这一个阶段,我们要做的是完善各个部件,让我们的车有完整的轮廓。

彭骏涵:因为这次比赛进场前要拆车,所以需要分配一下进场前每个人装车的任务。

郭琪:在此之前,我们再次确认一下人员的分工:关正负责抓球的装置;我负责轮胎的安装;戴子龙负责储球的装置;于佳冬负责将车组装起来;郭婧汝和李欣然负责传感器接线方面。

郭婧汝:好的,那话不多说,我们现在就开始第一天的研究吧。

## 2 开发过程

(说明：请详细记录每一位同学的开发工作；并把所有硬件的设计图纸附在本日志中。请附图片)

郭琪、李欣然、戴子龙:在寻迹区,会遇到一些障碍,我们对于怎么避开这些障碍进行了讨论,开始我们想到红外测距传感器、超声测距传感器、等多种类型的传感器。

超声波测距传感器的工作原理是已知超声波在空气中的传播速度,测量超声波在发射后遇到障碍物反射回来的时间,根据发射和接收时间差计算出发射点到障碍物的实际距离。

而红外测距传感器的工作原理是红外传感器的测距基本原理为发光管发出红外光,光敏接收管接收前方物体反射光,据此判断前方是否有障碍物。根据发射光的强弱可以判断物体的距离,它的原理是接收管接收的光强随反射物体的距离而变化的,距离近则反射光强,距离远则反射光弱。我们将超声波传感器和红外线传感器进行了比较,相比红外传感器,超声波传感器的设计模块系统更加简单方便,同时因为超声波的特性,在不同介质中的传出速度更为稳定,而红外避障模块的整体设计更为复杂,抗干扰能力也相对超声传感器来说更差,我们最终选择了超声传感器,主要原因是红外传感器只能检测车和障碍物的距离,而超声

波是锥形发射的，检测范围更大，不仅能检测出车和障碍物的距离还能检测出障碍物大概大小，这样，我们就能更加准确地判断出具体要绕多大的距离才能避开障碍物，而红外传感器无法做到这一点，所以我们最终在避障方面选择超声传感器。

### 3 调试过程

郭琪、戴子龙、李欣然：在救援区，为了检测墙壁、障碍物，我们在车前后的左右两端安装了触碰传感器。但是我们所选用的触碰传感器，可触碰的面积很小，而车前面需要检测的面积很大，只在左右两端各加一个触碰传感器对于我们的检测是远远不够的，关正同学提出了一个解决方案：在两个触碰传感器上连上一个联动装置(板)(如图所示)，这样就大大增加了检测面积，使车的整个前段都可以进行检测。





在比赛过程中，当我们将人质(球，下面全部用球来代替)全部放在斗中后，需要背对撤离台，将球全部倒入撤离台中，在我们测试的过程中，发现了因车的后面没有正对撤离台而导致用斗将球倒入撤离台的过程中出现偏差，球不能保证全部倒入撤离台中，我们针对这个问题进行了探讨，最后其中一个人提出了一个比较可靠地解决方法：利用车后面的两个触碰传感器，车后两端两个触碰传感器同时碰到撤离台壁时，就说明车的后面已经正对撤离台且已经靠紧，这样就不会在将球倒入撤离区时出现将球倒到外面的情况了。

#### 4 本次工作总结

今天我们对电子方面的一些重要元件进行了了解，同学们关于规则理解的更深了。

日志编号：1.1

活动日期：3.

参与队员：彭骏涵

---

### 1 讨论过程

(说明：请详细记录每位小组成员的发言，并通过草图的形式，记录每位同学的意见；请附图片)

邓少淳:今天我们几个负责救援区的程序员模拟了一下赛场上可能出现的情况,并在学校机器人组里前辈的帮助下写出了几条代码。

李欣然:那我们现在来看一下。

郭婧汝:好的。

### 2 开发过程

(说明：请详细记录每一位同学的开发工作；并把所有硬件的设计图纸附在本日志中。请附图片)

根据规则，进入救援区时们的位置是随机的，因此我们制定如下策略:利用车辆侧的超声传感器测定车离两边的墙的距离(超声探测范围比红外更大，大约为2m,且探测范围大)，

若左超生值大于 10000 且右超声值大于 10000，则判断为门在中间，若只有左超生值大于 10000 则判断为门在右边，若只有右超生值大于 10000 则判断为门在左边。当门在中间时，前冲右转，当门在左边时，直接右转，当门在右边时，直接前冲。

### 3 调试过程

在调试过程中，我们发现以下问题:当车绕场一周后重新回到门处，须执行死程序通过大门，但是门口处的人质未清理，因此，我们在之前程序的基础上，增加了固定程序前冲抓球，清理门口的球。代码如下：

```
if(sy>10000)
```

```
{
```

```
fzhua();

SetMotor(_MOTOR_z_, 0, 100);

SetMotor(_MOTOR_y_, 0, 100);

SetTenthS(15);

SetMotor(_MOTOR_z_, 1, 0);

SetMotor(_MOTOR_y_, 1, 0);

tzhua(2);

SetMotor(_MOTOR_z_, 2, 100);

SetMotor(_MOTOR_y_, 2, 100);

SetTenthS(14);

SetMotor(_MOTOR_z_, 1, 0);

SetMotor(_MOTOR_y_, 1, 0);

SetMotor(_MOTOR_z_, 0, 100);

SetMotor(_MOTOR_y_, 2, 100);

SetTenthS(10);

SetMotor(_MOTOR_z_, 2, 100);

SetMotor(_MOTOR_y_, 2, 100);

SetTenthS(3);

SetMotor(_MOTOR_z_, 1, 0);

SetMotor(_MOTOR_y_, 1, 0);

fzhua();
```

```
}
```

又发现如下问题:当门在中间时,可能紧挨着撤离台,因此需要检测,同时利用读取前超声的值(经检验为 12500 到 17500 之间)与爪子上的触碰状态判断。

```
t = GetSysTime();

    t0 = t;

    while(t-t0<100)

    {

        t = GetSysTime();

        zhua = GetTouchD(_ADEXTEND_claw_);

        SetMotor(_MOTOR_z_, 0, 98);

        SetMotor(_MOTOR_y_, 0, 100);

        if(zhua==1)

        {

            cltzt=1;

            SetMotor(_MOTOR_z_, 1, 0);

            SetMotor(_MOTOR_y_, 1, 0);

            break;

        }

    }

}
```

进场后,我们比较了三种搜人质方案(分别为巡墙绕场搜人质,沿场地中轴线直线行走同时检测人质,以及地毯式搜人质),认为巡墙绕场搜人质最为合理。因为沿场地中轴线直线行走同时检测人质难以确定车是否处于中轴线,且场地长宽不一,超声返回值不稳定,而且检测人质并抓取后难以再次返回中轴线。地毯式搜人质时间长,效率低,不方便处理障碍物,可能会撞上,不方便检测撤离台,且难以寻找大门并退场。巡墙绕场搜人质效率高,且对场边及墙角都有处理,便于利用车声右边超声巡墙,用车左边红外搜人质,全场无死角,因此我们采用这一方案。

进场有需要巡墙，我们利用车体右边的超声传感器巡墙。之前考虑用罗盘巡墙，但罗盘只能检测方向，不能检测离墙距离，而我们需要让车在固定距离处巡墙，因此我们选用能实时检测距离的超声传感器巡墙，利用 pid 算法，距离约为 3cm，因为这样使车前抓球的爪子张开放平后能将车右边的人质全部抓到，又同时防止当门在左边时经过门时车前触碰传感器碰墙误检。 if(sy<10000)

```
{  
  
    SetMotor(_MOTOR_z_, 0, speed1);  
  
    SetMotor(_MOTOR_y_, 0, speed1);  
  
    SetCentiS(8);  
  
    break;  
  
}  
  
    SetMotor(_MOTOR_z_, 0, 97);  
  
    SetMotor(_MOTOR_y_, 0, speed1);  
  
}  
  
    t_clt=0;  
  
}  
  
    sy>=sf+chayw  
  
    SetMotor(_MOTOR_z_, 0, speed1);  
  
        SetMotor(_MOTOR_y_, 0, speed2);  
  
    }  
  
    else if ( sy<=sf-chayn )  
  
    {  
  
        speed=rat2*abs(sz-sf)/100;  
  
        speed=speed2-speed;
```



```

        SetMotor(_MOTOR_z_, 0, 98);

        SetMotor(_MOTOR_y_, 0, speed1);
    }

    else
        /

    {

        speed=speed1- rat1/100*abs(sy-sy0);

        if ( speed>100 ) speed=100;

        if ( sy>=sy0 )

        {

            SetMotor(_MOTOR_z_, 0, speed1);

            SetMotor(_MOTOR_y_, 0, speed);

        }

        else

        {

            SetMotor(_MOTOR_z_, 0, 98);

            SetMotor(_MOTOR_y_, 0, speed1);

        }

        t0=t;

        sy0=sy;

    }

}

SetMotor(_MOTOR_z_, 1, 100);

SetMotor(_MOTOR_y_, 1, 100);

```

调试过程中，发现车体左右摇晃不稳，我们减小了左右两电机的差速度，降低了电机的初速度，从 60 减到 50，使车身稳定，沿直线行走，便于检测球与障碍物。

关于撤离台的处理，我们的策略是先检测撤离台，利用读取前超声的值与爪子右边上的触碰状态判断，若检测到撤离台，当爪子右边上的触碰检测到时，就向右转，若无，则直走，这样是车体逐渐走到撤离台平行的状态，此时左转，后退，立起车斗放下人质。再左转前进

爪子右边上的触碰检测到时，就向右转，若无，则直走，知道车体离开撤离台并与墙保持平行。

在调试过程中，我们发现，撤离台前面有一个三角形区域无法检测球的状态，而且当车在撤离台前左转时爪子可能将球碰开，并且后退时无法得知车后是否与撤离台平行，可能导致人质未放入撤离台内，我们的办法是再左转之前抬起前爪，左转后执行固定程序清理撤离台前的人质，后退时检测车后触碰传感器的状态，若两个传感器都检测到，则停止，否则继续后退，知道卡齐撤离台为止。

```
if ((squp<22000&&squp>17000&&zhua==1)||cltzt==1)
```

```
{
```

```
    t = GetSysTime();
```

```
    t0 = t;
```

```
    while (t-t0<300)
```

```
    {
```

```
        t = GetSysTime();
```

```
        qz = GetTouchD(_ADEXTEND_qz_);
```

```
        qy = GetTouchD(_ADEXTEND_qy_);
```

```
        zhua = GetTouchD(_ADEXTEND_claw_);
```

```
        if(zhua==1&&t-t0<150)
        {
            SetMotor(_MOTOR_z_, 2,100);
            SetMotor(_MOTOR_y_, 0, 100);
            SetCentiS(15);
        }
        else if (zhua==0&&t-t0<150)
        {
            SetMotor(_MOTOR_z_, 0,100);
            SetMotor(_MOTOR_y_, 0, 100);
        }
        else if(zhua==1&&t-t0>150)
        {
            tzhua(3);
            cltzt=0;
            break;
        }
    }
}
```

zhuan90z(90,zt1);

zq();

pengtai();

fqiu();

```

zhuang90y(100);

fzhua();

t = GetSysTime();

t0 = t;

while (t-t0<180)
{
    t = GetSysTime();

    qz = GetTouchD(_ADEXTEND_qz_);

    qy = GetTouchD(_ADEXTEND_qy_);

    zhua = GetTouchD(_ADEXTEND_claw_);

    hwz = GetIRDist(_IRDISTANCE_irz_);

    if(hwz>hwzf) cltqzt=1;

    if(zhua==1&& t-t0<180)
    {

        SetMotor(_MOTOR_z_, 2, 100);

        SetMotor(_MOTOR_y_, 0, 100);

        SetCentiS(15);

    }

    else if (zhua==0&& t-t0<180)
    {

        SetMotor(_MOTOR_z_, 0,100);

        SetMotor(_MOTOR_y_, 0,100);

    }
}

```

```

        SetMotor(_MOTOR_z_, 0,100);

        SetMotor(_MOTOR_y_, 0,100);

    }

}

```

检测人质方面，我们的策略是在车里右边加装一个能测定距离的传感器，传感器的探测范围约场地地宽度一半，当传感器值发生变化时说明检测到人质。我们决定使用红外传感器

因为它检测范围小，红外线为 15 度的光线，不容易误检，若使用超声传感器，由于超声检测范围大，人质体积过小，容易误检。当红外值大于某个值时，说明有人质在车左侧，此时车将前爪抬起，后退卡墙，放下爪子前冲抓起人质，加以时间保护。同时在两抓中间安装红外对射传感器监测抓内是否有人质。抓起后后退卡墙，右转 90 度，放下前爪子继续前进。

在调试过程中，我们发现，当红外传感器的返回值比较大时，说明人质离车较近，当车左转时容易碰到人质，导致救不到人质。因此，我们设计了如下方案：再定义一个较大的红外阈值当红外值大于这个值，车收起前方爪子，后退，左转一定距离，放下爪子前冲抓取人质，防止人质被碰开。

```

squp=  x3sou(_SOU2_squp_, 50000);

        hwz = GetIRDist(_IRDISTANCE_irz_);

        zhua = GetTouchD(_ADEXTEND_claw_);

        sy=  x3sou(_SOU2_sy_, 50000);

        if (squpzzt==0&&hwz>hwzf&&squp<squpz)

        {

            squpzzt=1;

        }

        if(hwz> hwzf&&squp>squpz)

```

```

{

    squpzzt=3;

}

if(squpzzt>=2

{

    zt1=2;

        if(squpzzt==3) tzhua(zt1);

    SetMotor(_MOTOR_z_, 0, 100);

    SetMotor(_MOTOR_y_, 0, 100);

    SetTenthS(2);

    zhuan90z(90,zt1);

    squp=  x3sou(_SOU2_squp_, 50000);

        if(squp<30000)

        {

            za(zazt);

            zazt=1;

        }

        else

        {

            zqz();

            zhuan90y(100);

            fzhua();

        }

    zt1=0;

```

```

        squpztt=0;

    }

    SetLCD5Char(150, 150, squpztt, YELLOW, BLACK);

    if (squpztt==1)

{

    squpz=  x3sou(_SOU2_squp_, 25000);

    }

    *

    //if (squp<squpq&&sy>8000)zt1=1;
    if ( squp<squpq )

{

        qjin(50);

        tzhua(1);

        if(squpztt==1)

        {

            squpztt=2;

        }

        else

        {

            qjin(150);

            touch1= qjin(150);

            if (touch1==1)

            {

```

```
        htui(60);

        fzhua();

    }

    else

    {

        zhuan90z(95,zt1);

        fzhua();

    }

    zt1=0;

    touch1=0;

}

SetLCD5Char(150, 150, squpzxt, YELLOW, BLACK);

}
```

#### 4 本次工作总结

这次是关于程序员救援区方面的研究，日后有待加强。



## 1 讨论过程

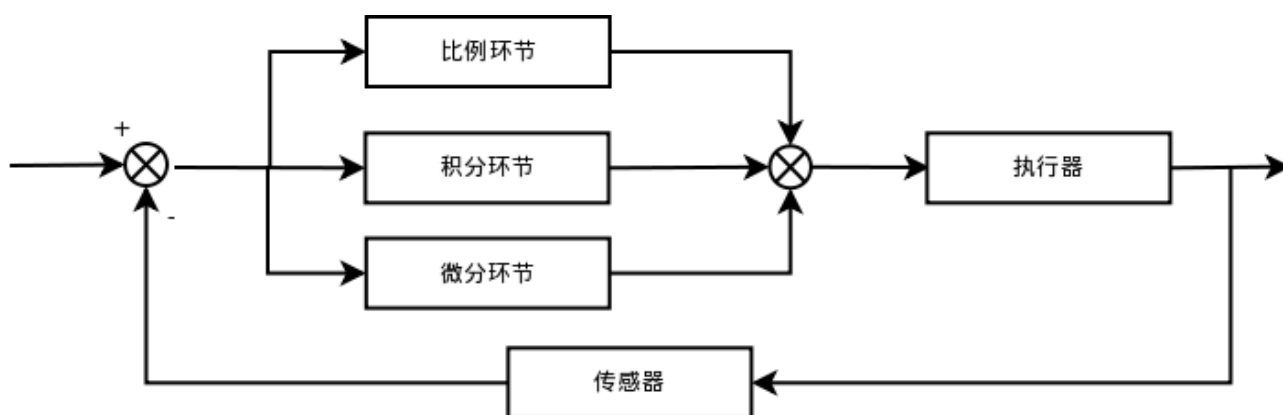
(说明：请详细记录每位小组成员的发言，并通过草图的形式，记录每位同学的意见；请附图片)

彭骏涵:我们负责走线的程序员今天也写了几条代码，在下面会对大家做出详细的讲解，希望大家能有所了解。

在我们的调试过程中我们发现采用碰拐的方式黑线循迹并不适合，会导致整体速度偏慢。因此我们决定选用另一种算法。我们使用了谷歌搜索引擎查询了有关控制算法，发现普遍用于工业生产上的 PID 算法可以很好的满足我们的要求。PID 算法，闭环控制算法，即 P 比例 I 积分 D 微分的结合。由于我们当前没有学习微积分的数学知识，组内程序员一起向数学老师询问微积分的相关知识。经过共同努力，我们学会了比例算法 P 的使用，并成功运用在了程序中取代碰拐。比例算法即设定一个阈值为标准值，当灰度传感器返回的值与阈值有偏差的时候，计算出误差值，乘一个比例系数，将得到的结果加在电机的转速上，以控制两电机转速一致，车辆直行。下图一为 PID 算法完整公式。图二为 PID 算法增量型公式。图三为闭环控制原理。

$$u(x) = kp(err(t) + \frac{1}{T} \cdot \int err(t) dt + \frac{T_D d err(t)}{dt})$$

$$u(k-1) = K_p(err(k-1) + K_i \sum err(j) + K_d(err(k-1) - err(k-2)))$$



在负责机械电子的同学把颜色传感器装上后，在我们的测试中我们发现，检测到绿色就转弯的程序并不可行。按照该方法，车连基本的走线都无法完成。加上显示屏功能后发现其在运行过程中随机误检测到绿色。经过与学长的沟通，了解到 RGB 色彩中，黑色往往会包含着绿色等色，导致颜色传感器的误检。于是在学长

的指导下使用了一段时间内读取检测次数的方式确定是否处于色块上方。(见方案一)经过了一番改动发现的确有很好的效果,然而又出现了新的问题——经常在同一色块上会连续触发多次色块检测,导致车辆偏离黑线,经过讨论决定采用保护程序——即在每一次色块程序运行后强制执行几秒的 PID 算法,使车辆离开色块范围。此种方法的确解决了多次色块误检的问题。在不断的测试过程中,我们发现如果色块放置的位置不在直角弯前,而是在弧形弯前,转弯 90° 的方式会导致车辆转弯角度过大偏离航线,结合硬件情况,我们决定将定角度转弯改为持续自转至中间光电检测到黑色停止。(见方案二)但是讨论后我们并没有发现处理线后色块的有效方式。

## 2 开发过程

(说明:请详细记录每一位同学的开发工作;并把所有硬件的设计图纸附在本日志中。请附图片)

方案一:

```
cz = GetColorSensor(_COLOR_left_, 1, 4);

cy = GetColorSensor(_COLOR_right_, 1, 4);

wz = GetADScable10(_SCABLEAD_wz_);

wy = GetADScable10(_SCABLEAD_wy_);

if(cz==1)

{

    SetMotor(_MOTOR_z_, ML, SL);

    SetMotor(_MOTOR_y_, MR, SR);

    if(wz<=1200 || wl<=1200);

    {

        SetMotor(_MOTOR_z_,2, 40);

        SetMotor(_MOTOR_y_, 0, 40);

        SetCentiS(125);

    }

}
```

```

}

if(cy==1)

{

    SetMotor(_MOTOR_z_, ML, SL);

    SetMotor(_MOTOR_y_, MR, SR);

    if(wz<=1200 || wl<=1200);

    {

        SetMotor(_MOTOR_z_,0, 40);

        SetMotor(_MOTOR_y_, 2, 40);

        SetCentiS(125);

    }

}

```

方案二：

```

SetLCDClear(63488);

    t0 = GetSysTimeMs();

    while(1)

        {

            SetMotor(_MOTOR_z_, 0, 100);

            SetMotor(_MOTOR_y_, 0, 100);

            t = GetSysTimeMs();

            if(t - t0>=200)    break;

        }

```

```

t0 = GetSysTimeMs();

while(1)

    {

SetMotor(_MOTOR_z_, 0, 100);

SetMotor(_MOTOR_y_, 2, 80);

t = GetSysTimeMs();

if(t - t0>=200)    break;

    }

```

```

while(1)

    {

SetMotor(_MOTOR_z_, 0, 100);

SetMotor(_MOTOR_y_, 2, 100);

ny = GetADScable10(_SCABLEAD_ny_);

zhong = GetADScable10(_SCABLEAD_zhong_);

if(ny<=nyf)    gy=1;

if(gy==1&&zhong<=zhongf)

    {

gy=0;

break;

    }

}

SetLCDClear(BLACK);

```

### 3 调试过程

#### 第一次 PID 尝试：

车辆在转弯处反复卡顿。经过我们小组同学的讨论，发现是我们为了使车辆在转弯时速度变快而将比例系数  $K_P$  设置过大造成的。比例系数过大导致了每次调整的误差值都很大，出现了超调现象。经过查阅资料(见下)我们发现，PID 算法三个常数项值的调整不可避免的需要使用经验调节法，这就导致了每次更换场地都需要手动调节常数项的值，这个问题有待于以后讨论。我们将常数项的值缩小后超调的问题被解决。

资料：.比例系数  $K_p$  的作用是加快系统的响应速度，提高系统的调节精度。 $K_p$  越大，系统的响应速度越快，系统的调节精度越高，但是容易产生超调，甚至会使系统不稳定。 $K_p$  取值过小，则会降低调节精度，使响应速度缓慢，从而延长调节时间，是系统静态、动态特性变差；

#### 第二次 PID 尝试：

成功完成基本循迹且效果良好，基本达到了时间要求。

#### 有关色块的尝试：

##### 第一次尝试：

由于出现颜色传感器误检的情况，循迹根本无法完成。百经思索也无法解决，通过向学长求助，得到了有关 RGB 色彩的提示，查阅了有关 RGB 的资料我们明白导致误检的原因，加以解决。

##### 第二次尝试：

由于没有写保护，导致连续检测同一个色块。有了之前的经验，很快我们就意识到了问题所在，加了保护程序解决了问题。

重要思路：屏显可以有效的帮助程序员查找错误，以后应尽可能多加入屏显！！！！

第三次尝试：

可以有效处理线前色块，但线后色块还没有思路，有待讨论。

#### 4 本次工作总结

走线时还是会有很多现场情况需要考虑，希望日后再程序方面的研究更加全面透彻。

日志编号：1.1

活动日期：3.

参与队员：彭骏涵

---

## 1 讨论过程

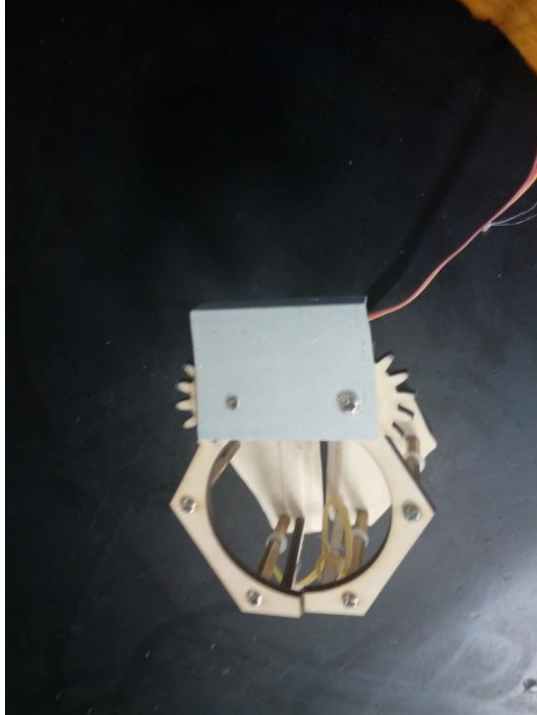
(说明：请详细记录每位小组成员的发言，并通过草图的形式，记录每位同学的意见；请附图片)

郭琪:今天打算设计一下抓球和储球的装置,大家有什么建议吗?

郭婧汝:我这里有几个设计出来的粗略的爪子模型,是大家集思广益后对抓球装置的理解的实体。

关正:那么今天我们就一起讨论一下抓球装置的选择吧。





## 2 开发过程

(说明：请详细记录每一位同学的开发工作；并把所有硬件的设计图纸附在本日志中。请附图片)

郭婧汝、关正、郭琪、于佳冬、戴子龙:接着上一阶段的研究试验,之前设计的收集球和运球为一体的装置模型做好后,经过联系场地的实验以及与规则要求各



项的结合，我发现虽然它在之前一次抓两个球的基础上对于抓球数量的确保和一次成功率有了很大的保障，但正因为这一装置大到可将五个球都容纳下，且它继续要满足容纳球数又要能够收集到球，所以对于这种结合一体的方案，在目前来讲，也只能“鱼和熊掌不可兼得”，即若要用这种大装置抓球，还是有些大材小用。这种救援装置的结构并不可取。

吸取之前两次设计成果的弊端，经过与关正、于佳冬讨论，我们认为在正式比赛场上，两种功能兼具的救援装置并不能很好的完成项目，因此我们试图将收集以及运输球这两种任务通过两种装置的形式来分别体现其功能。所谓灵感来源于生活，我们假设把司空见惯的抓娃娃机当作救援场地，那么娃娃就相当于场上的“人质”。我们便可以把其中负责抓取的机械爪当作场上收集球的装置，而我们还能发现的是抓娃娃机每次都是将抓到的物品，放到固定位置，以此及彼，我们想是否也可采用异曲同工的方法来完成救援“人质”的任务。所以我们想尝试将收集装置作成和前者提到的抓娃娃机中一样收集抓取功能的爪子，然后与前者相同，我们想设计一个固定装“人质”来运输球到放球区的装球筐。

在有了初步关于把两者功能分别体现的想法且结合实际做出决策之后，我与关正、于佳冬、郭琪结合比赛规则讨论，将两种装置分组，即在两人主要攻克一种装置的体现方式时，另外两人可提些建议。首先对于爪子的部分：我喝郭琪想用一些简单机械结构来做到爪子的张开和合拢，一开始先想到的是可否使用之前提及的抓娃娃的爪子一样的爪子来收集到球，但通过与程序员的讨论，我们发现若要做到和抓娃娃机一样的爪子对于程序上难度较大，且抓取的精准度显然不够。所以我们还是决定采用爪子在水平面张开然后救援“人质”的方法。于是我们想到用一些简单的机械结构来完成爪子的张开与合拢的效果，且因为改为水平面上开合的爪子，我们倾向于乐高的简单机械爪的左右各两个完全一样的弧形挡爪构成的一组爪子。（通过物理上的学习，我们在学习线速度角速度时接触过齿轮，所以比较侧重于这种原理）所以我们决定先采用齿轮原理制作爪子，为了想尽量减少舵机使用，我尝试用一个小齿轮带动两个较大齿轮旋转，且通过爪子固定在较大齿轮上，来实现小齿轮不同方向的转动带动爪子的两边打开或合拢。

在爪子部分有了初步想法的同时，关正和于佳冬的装球筐部分也有了一定的初步进展，经一起协作讨论，我们首先想不用舵机带动筐的翻转来实现“人质”可以顺利的放入放球区。而结合规则，我们先确定了筐的规格大概为一个可装最多六个球的矩形小筐，因之前在考虑两种功能结合一体的装置时对于筐的挡板已经有了可以确定的高度尺寸，即大约为 15mm 的高度。经过查找资料，想在这一部分使用 双稳态 的原理（双稳态开关意为具有两种稳定状态，当接受的触发信号后由初始稳定状态变为另一种稳定状态，当再次接受到触发信号后，装置稳定状态又会变成初始的稳定状态）而对于如何实现这一想法还需要进一步实践。

爪子在有了虚拟构图后，(inventer 上画齿轮啮合的步骤:选中齿轮上一个面，将其约束在另一齿轮上对应的平面)之后运用激光切割将草图上部件做出，通过螺丝连接，制作出木制模型，并将舵机连接到爪子上，但发现了一个巨大的漏洞就是如果这样安装，势必要使连接小齿轮的舵机跟着爪子一起进行运动，这样说明，我们反而没有减少舵机的使用数量，反而对于下方带动整个爪部竖直切面上进行角度上改变的舵机有了更大的要求。而且多个齿轮的连锁带动，在此时也显得有些过于繁琐。

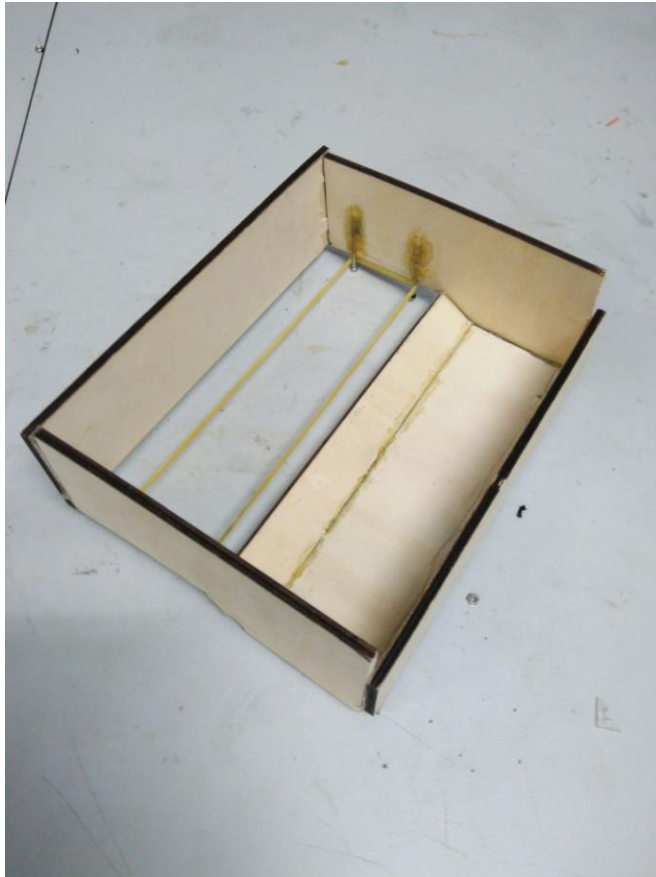
装球装置的想法同时开始进行，我们先用木板做出了除后挡板之外所有的结构，然后对于后挡板，为了实现之前所说的原理，我们决定在后挡板下部分多出一块，这样就可以实现在力的作用下，装置从一种状态变为另一状态的程度，之后经过不断探讨实践，我们又在后挡板的外部用热熔胶再粘一块木料。这样可以确保原理的实现。

但之前做的装球装置在经过操作实验时发现，其正因为有两块额外木料的所以在之前的越障任务时有很大的不便，于是因为这种设计的不便利，弊大于利，并不准备采用此种方式。

### 3 调试过程

爪子方面我们又想到了之前用皮筋的想法，只是在现在的爪子想法中，我们不把皮筋当作主要，而是辅助爪子的抓取，来让爪子抓到球后不会因为爪子碰球而让球弹走。而对于如何实现爪子识别到球，我们根据规则，得知“人质”可反光，因此我们可以利用红外对射来判断球是否在爪子中间，而皮筋的作用就是在抓球时，能利用其弹性，将球很好的抓住，不至于跑出。之后我们做草图，制作模型，因准备只用两个舵机分别带动两侧爪子部件，我们便连接上舵机进行简单的抓球实验，发现结果相对于之前几种设计都有显著提高。

装球装置通过之前的实验发现矩形斗还是过于占地，且考虑到车的主体部分，我们想设计一个斗装的装置，这样可以和罗盘架等结构较完美的契合。并且在前端有和之前设计的筐和皮筋结合一样的斜坡，可以让球向后走。且整体斗部使用舵机带动来使斗向后倾斜，进而将球放入放球区。经过试验后发现，这种斗的形式对于比赛还比较便利。至此，对于负责分别收集和运输的“人质”的装置(斗和爪子)，都有了具体的结构，而下一步要做的就是对于这两种装置更为细致的改动。



#### 4 本次工作总结

今天对于抓球装置和储球装置的研究很深入, 这个方面已经达到了比较成熟的阶段。

日志编号：1.1

活动日期：3.

参与队员：彭骏涵

### 1 讨论过程

(说明：请详细记录每位小组成员的发言，并通过草图的形式，记录每位同学的意见；请附图片)

关正：今天进入收尾工作了，对于车的大体安装已经有一个明确的方向了。

戴子龙：是啊，好的开始好的结束，加油。

### 2 开发过程

(说明：请详细记录每一位同学的开发工作；并把所有硬件的设计图纸附在本日志中。请附图片)

由于车上有很多传感器，还需要装电机与舵机等元件，所以需要有一个强有力的车板，不仅能承载各种原件，也需要轻便以适合车的行进。根据之前比赛的设计，进行修改。在循迹过程中，车要检测障碍并绕过障碍，所以在车的左右和前面要安装超声传感器；因为要循迹，所以要在车板下靠近前边安装一个灰度传感器；因为要在比赛中确定方向，所以要在车的罗盘架上安装罗盘。因为车身高度的原因，所以爪子的舵机和斗的舵机要固定在车板的上面；经过讨论过后，决定把稳压板固定在车板上面，驱动板固定在车板下面，旁边的空间还可以固定电池。

### 3 调试过程

对于控制器，讨论后决定安在斗的舵机上面，也可以借助斗的舵机支撑。今天与程序员讨论，在救援区车后面需要触碰传感器来确定撤离台的位置，所以决定在车板下两侧分别安装两个触碰传感器。而在循迹区和救援区都需要车体前方有触碰传感器来检测障碍物等，然而前触碰又无法实行大面积的检测；在讨论中，关正想出了解决方案——在两个前触碰前加装一个联动装置，可以增加车前检测范围，为了防止在救援区车体前触碰直接碰到墙壁而损坏前触碰板亦或是前触碰，戴子龙提出在触碰板两侧加装两个导轮来防止此现象发生。最后，确定了在车板上都需要安装的元件：一个罗盘(安装在铝架上)、一个用来装斗的后舵机、一个用来装爪子的前舵机、一个控制器、一个驱动板(车板下)、一个稳压板(车板上)、一个用来固定超声的铝架、四个用来固定轮胎的铝架、用来固定红外传感器的铝架、装在车前的两个触碰传感器(及联动板)和装在车后的两个触碰传感器。

### 4 本次工作总结

第二阶段的研究也算是圆满完成，我们现在已经有一个车的大体框架了，在接下来就可以对车体进行细节的改动了。