

## 一种基于红外循迹传感器的 PID 循迹算法

红外传感器循迹原理如图 1 所示：

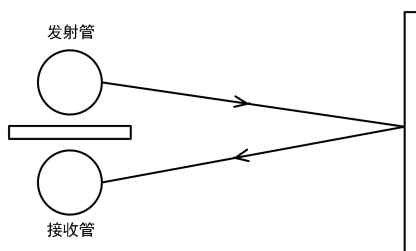


图 1 红外传感器原理

红外二极管发射红外光，接收管接受反射的红外光信号。不同的颜色反光效果不同，如果红外光照射在黑色物体上，由于黑色物体对光的吸收能力强，反射的光很少。但照在白色物体上，由于白色物体对光的吸收能力弱，反射的光较多。从而当红外二极管发出的光照射在不同颜色的物体上时，接收管接受的红外光强度也不同，从而可以判断出前面物体的颜色。



图 2 赛道

红外循迹就是根据这个原理，将红外模块安装在寻迹小车上，然后再赛道上面贴上黑色胶带，当检测到黑色时说明小车在黑线上，检测到白色是说明小车偏离黑线，则需要向黑线靠近，赛道如图 2 所示。市面上大多数红外循迹传感器采用数字式的，也就是将红外接收管接受到的模拟量信号通过电压比较转换为 0 和 1 的数值信号，按照这种方法寻线的话至少需要两个红外传感器，固定在黑线两侧，当左边红外传感器检测到偏离黑线时往右偏，当右边红外传感器检测到偏离黑线时往左偏。这样确实可以寻线，但是寻线效果相当不好，这样寻线有以下几个缺点：

- ①寻线精度差
- ②容易偏离轨迹
- ③寻线速度慢

不过也有优点，优点就是程序简单。市场上也有五路的和七路的但是这样效果也就好那么一点。其实市场上这种将模拟量变成数字量的方法本身就将精度很高的传感器变为精度低的了。因为采集红外接收管信号的方式是通过 AD 采集，如果是 10 位 AD 的话精度位 1024，采用 STM32F103C8T6 的话有 12 位 AD，采集的精度位 4096。但这种采集精度高的处理方法是比较好，但是处理的过程相对麻烦，大致思路就是单片机通过内部 AD 采集循迹传感器模拟量，然后通过 PID 参数控制舵机角度。

下面我要介绍的红外循迹模块也就是这种直接通过单片机内部 AD 采集红外循迹传感器数据，然后根据数据 PID 动态调节舵机角度的方法。采用的红外循迹模块为三路循迹模块实物如图 3 所示，电路原理图如图 4 所示。



图 3 三路红外循迹传感器实物图

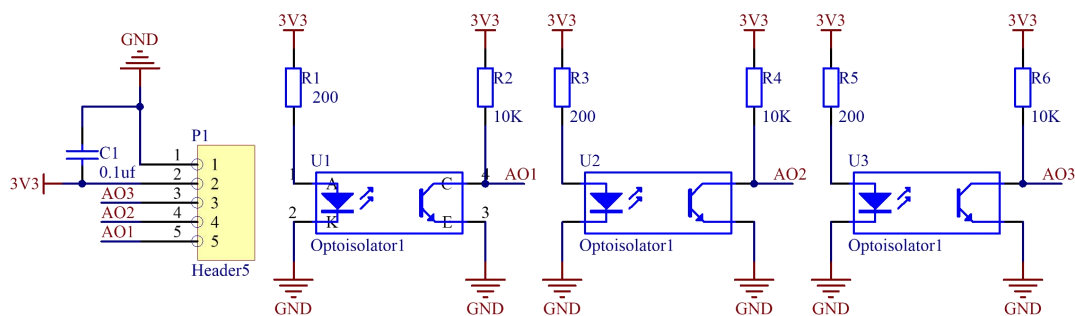


图 4 循迹模块电路原理图

其中三个红外管的功能不同，左右两个红外管的作用是判断传感器离黑线边界的距离，获取的 AD 值越大说明离黑线越远，则舵机需要转动的角度也就越大。中间的一个红外管的作用是判断是否传感器在黑线内，我将用一个曲线图来向大家表示原理，便于大家理解，图 5 位循迹模块安装图。

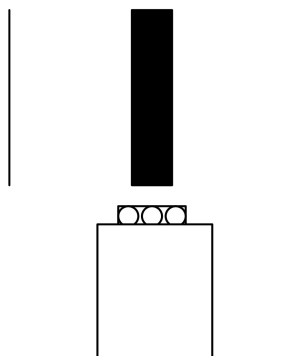


图 5 循迹传感器安装图

将循迹传感器固定在小车上，传感器离轨道的高度为 3cm 左右，假设左两个红外传感器之间的距离和轨道黑线的宽度相同为 3cm，传感器从左到右移动获得的图像如图 6 所示：

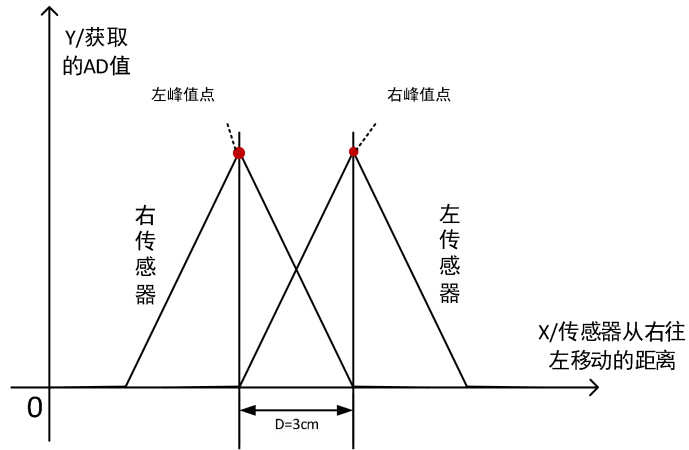


图 6 左右传感器向左移动图像

将左传感器值减去右传感器值可得到图像如图 7 所示：

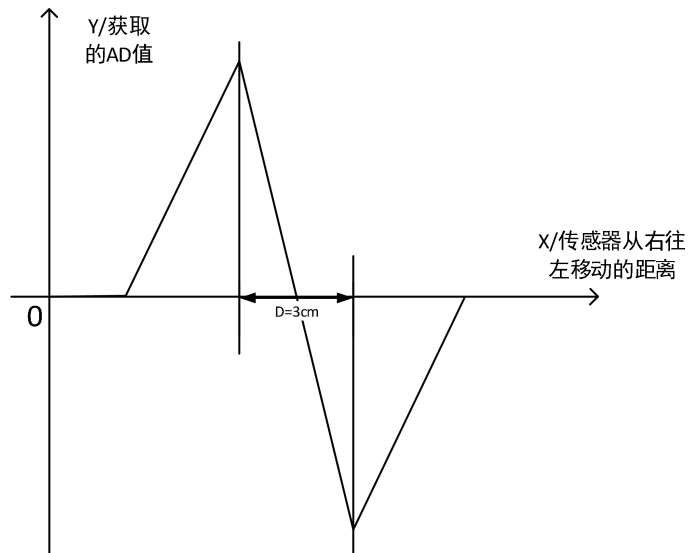


图 7 左右传感器相减数据整合图

这时获取的数据有效长度只能在 3cm 黑线轨迹之内，当数据为 0 时表示小车循迹方向没有偏差，大于 0 说明小车往左偏，小于 0 说明小车往右偏，数值越大说明偏离的越大。但是由于数据有效长度只有 3cm 当小车速度快时，容易冲出有效循迹黑线范围，同时我们也注意到，有效长度左边和右边有一段数据没有用到，我可以尝试将左有两段没用用到的数据纳入到有效处理数据。

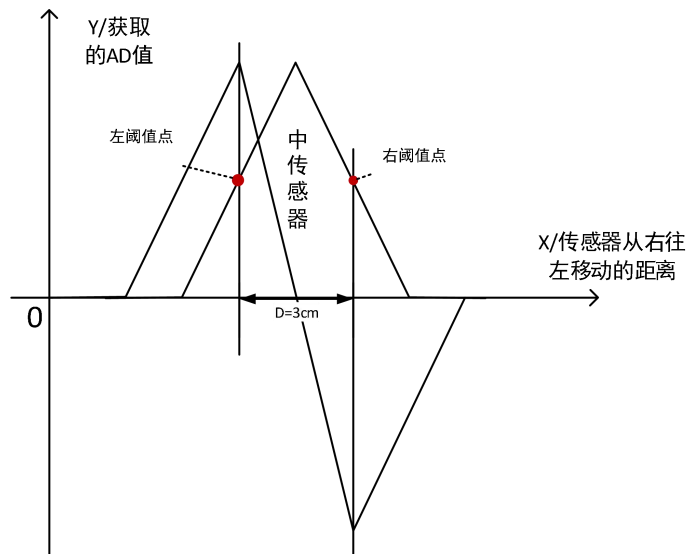


图 8 根据中传感器获取左右阈值点

如上图 8 锁死，我们可以通过中间的红外传感器来获取，当中间传感器小于阈值时，通过判断左右两个红外传感器相减值得正负来处理左右两边的数据，数据翻转将斜率变为原来 2 倍。得到的最终效果图如下如所示，最终得到的宽度为原来 3 倍左右，为 9cm。即使传感器完全偏离黑线，最终也会回来，最终达到处理后的效果为图 9 所示。

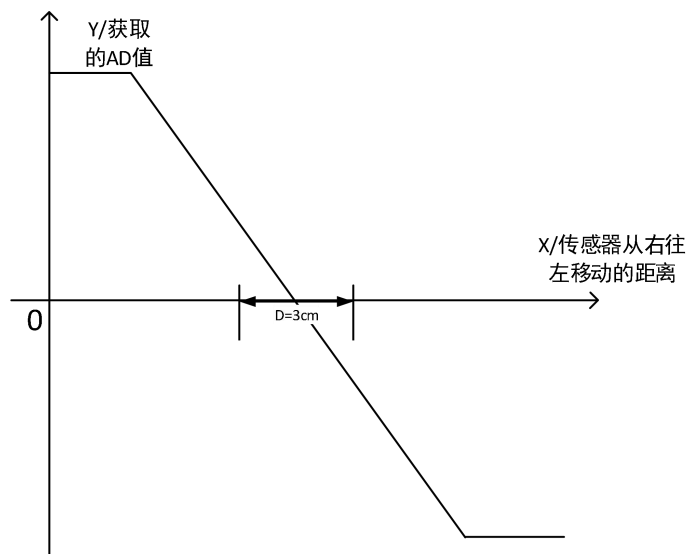


图 9 最终数据处理效果

综上所述，需要经过这几步达到最终效果：

(1)确定左右传感器差值(D\_AD\_VALUE):

确定方法：将传感器放到白色纸上，将左右两个传感器的数据通过串口发送到电脑上，获取左右两个传感器差值。

(2)确定左传感器峰值点(LEFT\_MAX)和右传感器峰值点(RIGHT\_MAX):

确定方法：将左红外传感器和右红外传感器分别从左到右移动，将数据通过串口发送到电脑上观察数据获取最大值，一般当传感器移动到黑线中间是获得数据最大值。

(3)确定左阈值点(LEFT\_THREASH)和右阈值点(RIGHT\_THREASH):

确定方法：当达到左传感器达到峰值时，观察中间红外传感器数值，此时中间红外传感器数值即为左阈值点。同样当达到右传感器达到峰值时，观察中间红外传感器数值，此时中间红外传感器数值即为右阈值点。

(4) 进行数据整合：

①当中间传感器获取的 AD 数值在左阈值点和右阈值点之间时，将获取到的左传感器数据 (Left\_AD) 与右传感器数据 (Right\_AD) 相减加上左右传感器差  $Data\_Out=(Left\_AD-Right\_AD+D\_AD\_VALUE)$ 。

②当①中获取的数据大于零( $DATA\_OUT>0$ )且中间传感器获取的数据小于左阈值时( $Mid\_AD<LEFT\_THREASH$ )，  $Data\_Out=(2*LEFT\_MAX-Left\_AD)*2$

③当①中获取的数据大于零( $DATA\_OUT>0$ )且中间传感器数据获取的数据小于左阈值时( $Mid\_AD<RIGHT\_THREASH$ )，  $Data\_Out=(2*RIGHT\_MAX-RIGHT\_AD)*2$

综上所述换成用 C 语言形式写即为：

//获取循迹传感器输出函数

//返回值：int 类型，返回循迹传感器数据，根据此数据调节小车转向

int GetTraceDate(void)

```
{
    Data_Out=(Left_AD-Right_AD+D_AD_VALUE);
    If((Data_Out>0)&&((Mid_AD<LEFT_THREASH))
    {
        Data_Out=(2*LEFT_MAX-Left_AD)*2
    }
    else if((Data_Out<0)&&((Mid_AD<RIGHT_THREASH))
    {
        Data_Out=(2*RIGHT_MAX-RIGHT_AD)*2
    }
}
```

注意：这是最理想情况下的调节步骤，在调试时需要根据自己的模块安装位置和黑线宽度调节一些参数，但大致调试步骤就是这样。