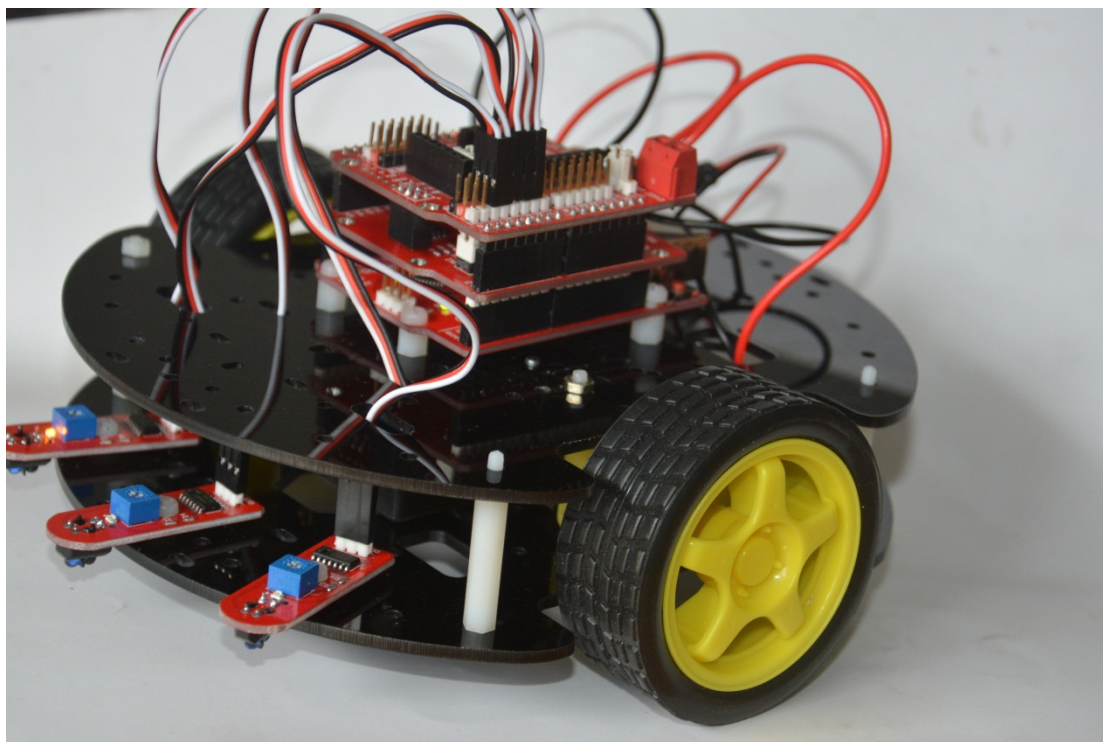


寻线小车

一. 概述

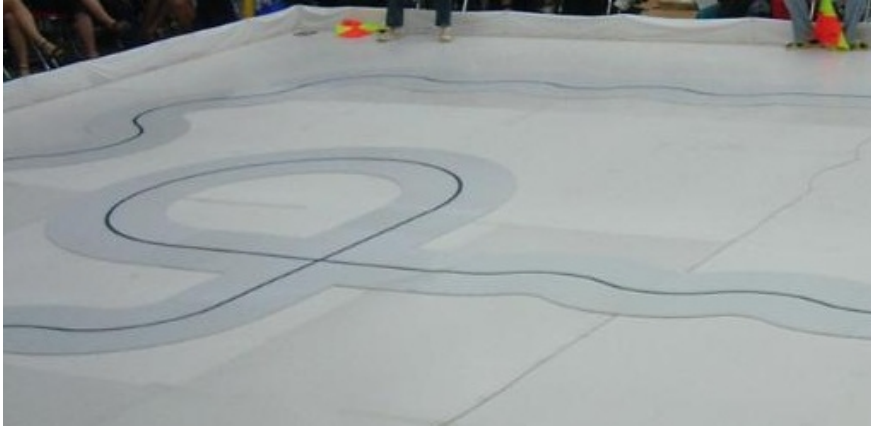
简介



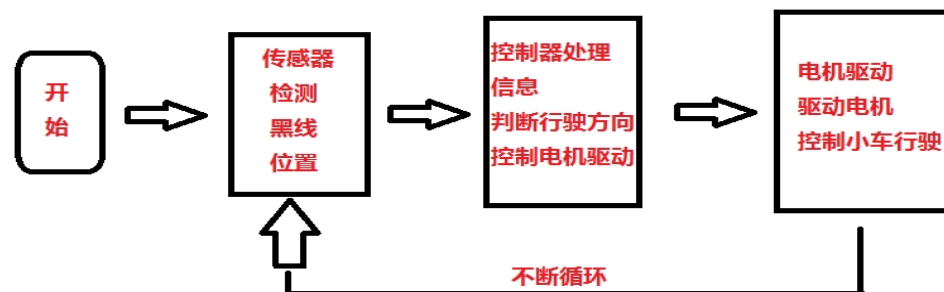
寻线小车作为智能小车制作的初级项目，因其组装试调简单，且涵盖了软硬件组装试调 and 智能控制思想等多方面的知识而深受广大爱好者和学习者们的喜爱。许多高校也常年开展寻线小车的比赛来训练学生的动手实践能力。通过制作寻线小车，我们可以学会控制器的基本原理和使用方法，红外传感器的特性和使用，如何使用电机驱动驱动电机，小车自动控制的原理和程序的编写等知识。且在制作寻线小车的基础上，往后还可以通过改装实现壁障，遥控，避崖，走迷宫等多种功能。

基本原理

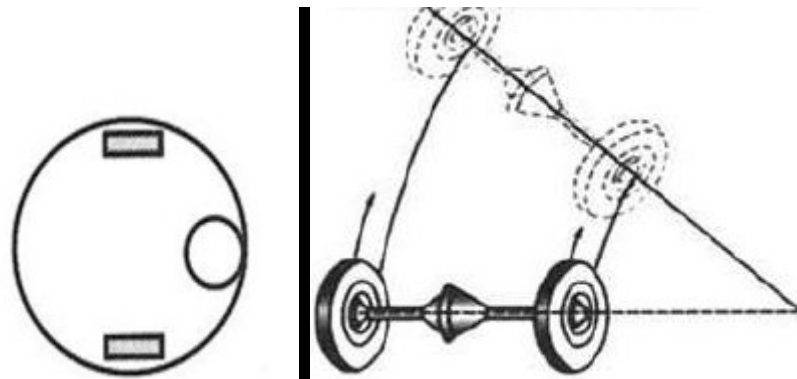
场地：寻线小车，顾名思义就是一台可以自己寻着线条行驶的智能车。跑道一般为白底黑线，黑线一般为电工胶带，白底用白色泡沫板或者浅色的地板。



控制流程：小车的红外传感器检测到黑线的位置后发送给控制器，控制器判断应该如何行走，再通过电机驱动控制电机转速来改变小车的行驶方向。



转向原理：小车有 2 个驱动轮和一个万向轮，可以用差速原理来控制小车转向。差速转向就是通过驱动两个轮子以不同速度运行，从而达到转弯的目的。例如控制你的小车左边电机转速慢，右边电机转速快，单位时间右轮行驶距离比左轮大，小车就自然往左边转动。目前大型的挖掘机、推土机、坦克都是采用这种驱动方式。由于驱动简单，对硬件要求的考虑，此次我们演示制作智能小车也采用这种驱动方式。



二．需要的材料

1. 车体：Openjumper 小车车体套件×1
2. 控制器：Arduino uno×1
3. 电机驱动：openjumper L298 电机驱动板×1
4. 扩展板：UNO 配套的传感器扩展板×1

5. 寻线模块：红外寻线传感器×3
6. 尼龙柱，螺丝螺帽，杜邦线，传感器 3p 杜邦线若干。

三. 硬件的组装试调

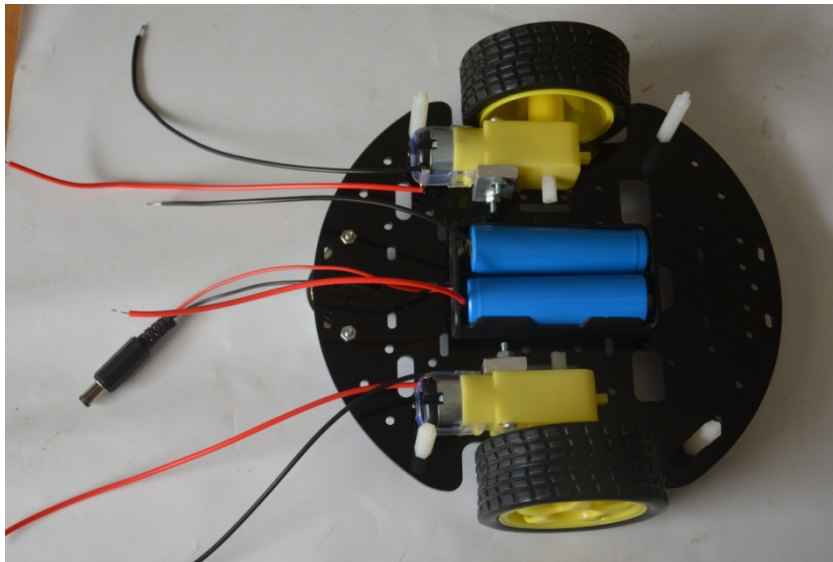
1.车体组装

详见小车车体组装教程。

注意事项：

1.先不要将上层车体板用螺母固定，组装板卡时还需要将上层板取下。

2.电机驱动板上的 VIN,GND 接线端子需要直接通过电池供电，以提供驱动电机的电流。所以在完成车体组装前最好先用 2 根电源线将电池的正负极引出备用。(小车套件不配电池，电池盒需要 2 节 18650 锂电池，需要单独购买。)

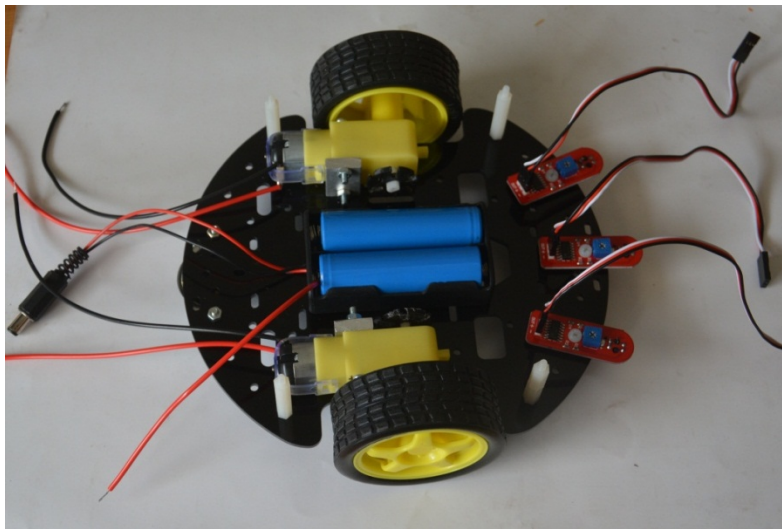


2.安装传感器



传感器的安装较为简单，用螺母，螺丝固定到小车底板前端的孔洞上即可。再将传感器

接线穿过上层板预留的孔后接到扩展板上。



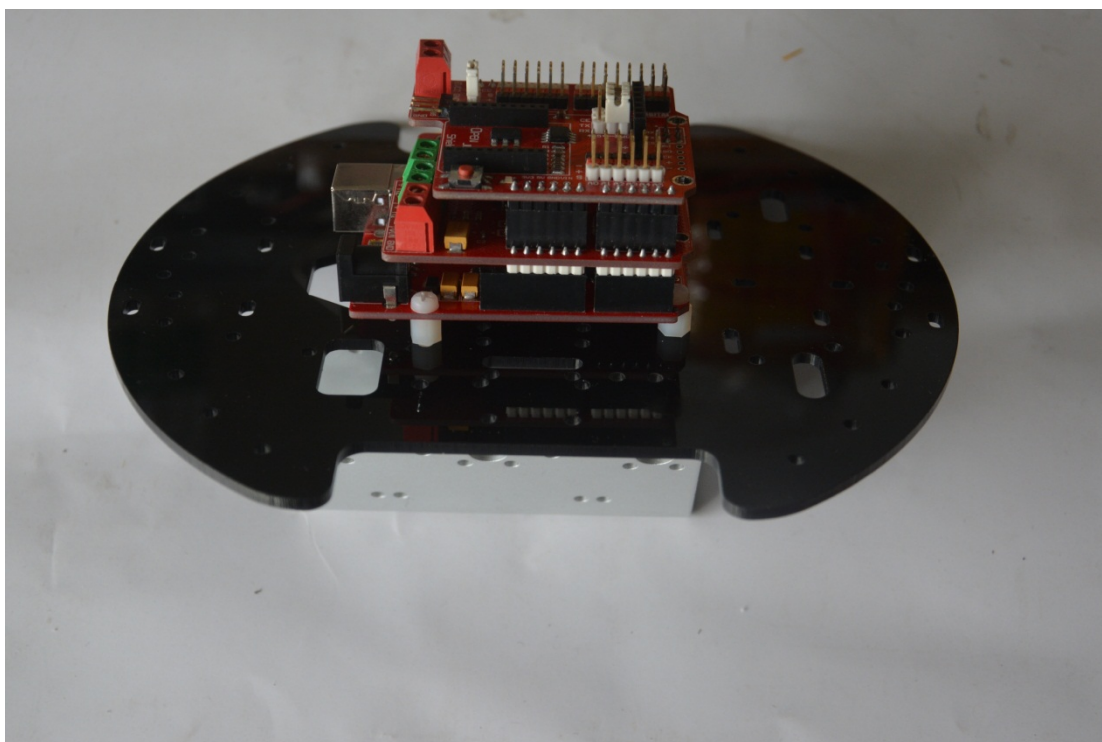
注意传感器线的使用：openjumper 的传感器专用 3p 杜邦线黑色为 GND 红色为 VCC 白色为信号线接 IO 口上，按颜色插在扩展板上即可。

3.板卡组装



如图，先将 arduino UNO 用尼龙柱或铜柱固定在小车的上层板。小车车体板上有与 UNO 兼容的孔洞，并且 2 个方向都可以安装，建议用图片上相同的位置安装，以便接下来电机和电源的接线。

用螺丝螺母固定好板卡后，将电机驱动板和传感器扩展板依次叠插在 uno 上，注意方向和针脚的对应。

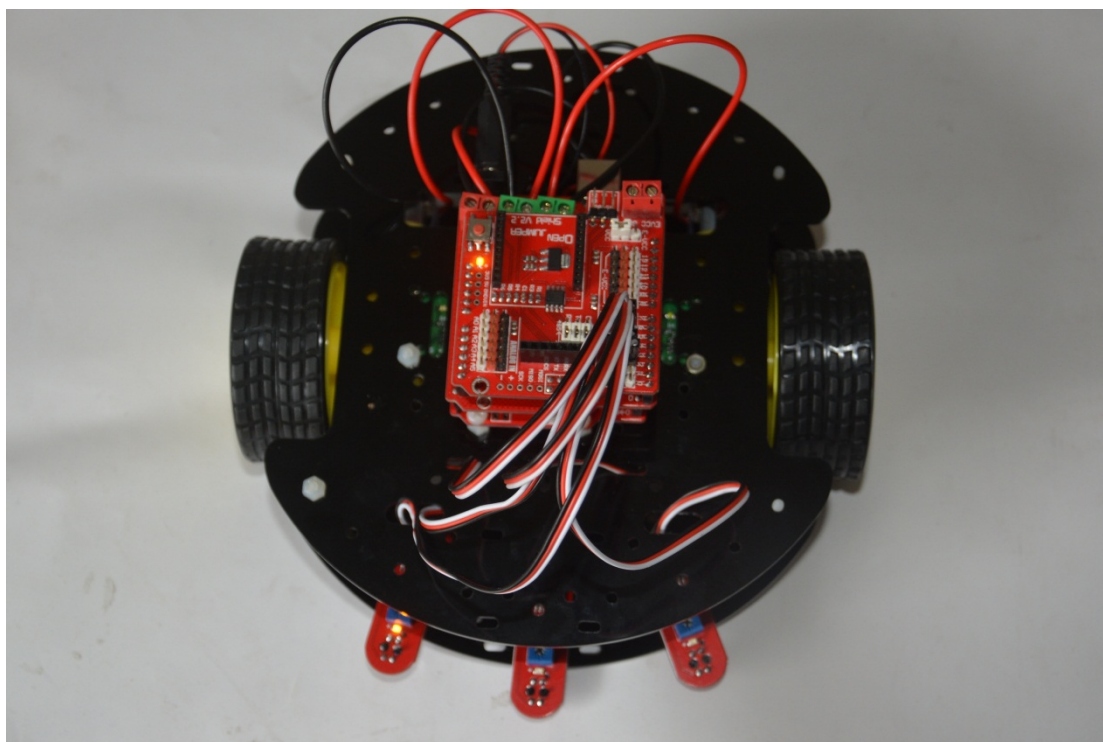


接下来就可以进行连线 and 电机的测试了。

将下层的电机线，电源插头，电机驱动的供电线穿过上层板的孔洞。

Uno 插上电源插头，通电正常后 LED 电源指示灯点亮。

将电池正极线连接到电机驱动板 VIN 负极接到 GND，供电部分连线完成。下面先下载一个程序来为电机接线做准备。



电机的 2 根线并没有正负之分，电机驱动板上的 M1,M2 接线端子也不能确定应该对应左右哪一个电机

将以下这个程序下载到 uno 中可以方便我们对小车电机进行连线

```

*
/*
电机测试程序
功能：测试电机驱动功能和电机连线是否正确
效果：左电机反转，右电机正转，电机转速由小到大。
*/
int INA = 4; //电机 A 正反转控制端
int PWMA = 5; //电机 A 调速端
int INB = 7; //电机 B 正反转控制端
int PWMB = 6; //电机 B 调速端
void setup(){
pinMode(INA,OUTPUT);
pinMode(INB,OUTPUT); //配置电机输出 IO 口为输出
}
void loop(){
int value;
for(value = 20 ; value <= 255; value+=2){ //循环每次速度增加 2 最大到 255
motosp(-value,value);//电机速度赋值，左电机负值为反转，右电机正转。
delay(50);
}
}
void motosp(int sp1,int sp2)//声明电机速度控制函数。括号内定义的变量分别为左右电机速
//度值，范围-255~+255，正值为正转，负值为反转。
{
if(sp1>0) {
digitalWrite(INA, HIGH);
}
else
{
digitalWrite(INA, LOW);
}
if(sp2>0)
{
digitalWrite(INB, HIGH);
}
else
{
digitalWrite(INB, LOW);
}
analogWrite(PWMA,abs (sp1));
analogWrite(PWMB,abs (sp2));
}
*

```

程序下载好后，将电机连线进行试接。连线正确后左电机反转，右电机正转，如果电机转向

错误，交换连线直到正确为止。

四．规格

控制器	Arduino uno
电机驱动	L298
驱动部分工作电流	2A
驱动部分输入电压	4.8～25V
逻辑部分输入电压	5V
长	200mm
宽	180mm
高	110mm
重量	520g
电机型号	130
电机电压	3-6v
转速	125 转/分
减速比	48/1
最大行驶速度	47.7 米/分
电机空载电流	110ma
电机堵转电流	150ma
车轮直径	65mm
底盘离地面高度	20-22mm

．引脚

I/O 口	类型	作用	描述
D4	数字输出	电机 A 正反转控制端	1 为正转 0 为反转
D5	PWM 输出	电机 A 调速端	电机速度 0-255
D6	PWM 输出	电机 B 调速端	电机速度 0-255
D7	数字输出	电机 B 正反转控制端	1 为正转 0 为反转
D11	数字输出	左传感器	检测到黑线为 0 未检测到为 1
D12	数字输出	中间传感器	检测到黑线为 0 未检测到为 1
D13	数字输出	右传感器	检测到黑线为 0 未检测到为 1

六. 示例代码

```
int SNUM[3] ;//定义 3 个传感器
int INA = 4; //电机 A 正反转控制端
int PWMA = 5; //电机 A 调速端
int INB = 7; //电机 B 正反转控制端
int PWMB = 6; //电机 B 调速端
void motospd(int sp1,int sp2);
void setup(){
  pinMode(2, INPUT); //配置 6 个传感器 IO 口为输入
  pinMode(3, INPUT);
  pinMode(8, INPUT);
  pinMode(11, INPUT);
  pinMode(12, INPUT);
  pinMode(13, INPUT);
  Serial.begin(9600); //打开串口设置波特率
  pinMode(INA,OUTPUT);
  pinMode(INB,OUTPUT); //配置电机输出 IO 口为输出
}
void loop(){
  SNUM[0] = digitalRead(2); //SNUM[0]-SNUM[2]分别为从左到右三个红外寻线传感器
  SNUM[1] = digitalRead(3);
  SNUM[2] = digitalRead(8);
  for (int i=0;i<3;i++)
  {
    Serial.print(SNUM[i]);
  }
  Serial.println(SNUM[3]); //用串口依次输出 6 个传感器的值 0 或 1

  if ((SNUM[0]==0)&&SNUM[1]&&SNUM[2])//左端传感器检测到黑线
  {
    motospd(50,120); //设置左右电机转速，使小车向左行驶
  }
  if (SNUM[0]&&(SNUM[1]==0)&&SNUM[2])//中间传感器检测到黑线
  {
    motospd(100,100); //小车直行
  }
  if (SNUM[0]&&SNUM[1]&&(SNUM[2]==0))//右端传感器检测到黑线
  {
    motospd(120,50); //小车向右行驶
  }
  delay(50);
}
```



```

void motospd(int sp1,int sp2)//电机速度控制函数。括号内分别为左右电机速度值，
{
    if(sp1>0)                //范围-255~+255，正值为正转，负值为反转。
        digitalWrite(INA, HIGH);
    else
        digitalWrite(INA, LOW);
    if(sp2>0)
        digitalWrite(INB, HIGH);
    else
        digitalWrite(INB, LOW);
    analogWrite(PWMA,abs (sp1));
    analogWrite(PWMB,abs (sp2));
}

```

七. 改进思路

虽然小车原理非常简单，但要让它跑的又快又稳是需要费一番心思的哦。本篇的教程讲述了一个最简单的寻线小车的制作方法，还有许多不足之处就需要大家自己去发现和改进了。下面给大家一些改进加强提示。

1. 加装测速码盘：测出小车 2 个电机的转速，可以对小车行驶状态精确控制。
2. 增加传感器数量：本教程只用了 3 个红外传感器，如果想让小车跑的更平稳，可以加装更多的传感器。
3. 使用其他的传感器：arduino 提供有数个模拟输入 IO，例如我们这里使用的 uno 板就有 6 个，分别为 A0-A5。所以我们可以使用模拟量的传感器，例如灰度传感器，每个传感器距离黑线位置不同时，返回的值为 0-1024 而我们用的传感器返回值只有 0/1 两个状态。用模拟量的传感器就能更精确的计算黑线相对小车的位置。
4. 改进算法：本篇教程的示例程序用的是最简单的判断算法。若要让小车跑的更好，可以改进小车程序，让它变得更聪明。例如自动控制中常用的 PID 算法。