

学号: 202010311229



上海海事大学
Shanghai Maritime University

《计算机硬件课程设计》报告

学 院: 信息工程学院

专 业 名 称: 计算机科学与技术

班 级: 计算机 203

姓 名: 孙思进

指 导 教 师: 张琳

2021 年 12 月 23 日

摘 要

为研究设计智能寻迹避障小车的控制部分,文中通过研究超声波模块,红外线模块完成小车的控制。首先通过 Arduino 平台以及 Arduino IDE 软件进行程序编程来控制小车的行驶。

寻迹小车由 MSP430 单片机模块、升压模块、电机驱动模块、红外光电传感器等部分组成。小车能够沿着地面上的长宽为 1 cm 的黑白相间方格行驶,并且可以根据方格路径实现小车的自动寻迹功能,小车在转弯处可自动减慢速度,以避免由于速度过快而偏离给定路线。当红外光电传感器阵列遇到长度为 5 cm,宽度为 1cm 的黑色条纹的时候,小车停车。

经过实验测试,该自动寻迹小车能够较好的实现寻迹功能,可被用在货物运输、实验教学和玩具中。

关键词: 寻迹小车; 红外线; 单片机

Key words: Trace trolley; Infrared; Microcontroller

目 录

摘要	I
1 任务安排	1
2 相关硬件说明	2
2.1 Arduino 单片机	2
2.1.1 Arduino Uno 介绍	2
2.1.2 引脚介绍	2
2.1.3 端口介绍	3
2.1.4 供电方式介绍	4
2.2 红外收发模块	4
2.2.1 模数转换	4
2.2.2 数字端口介绍	5
2.3 L298N 介绍	5
3 小车结构与系统模块设计	7
3.1 小车结构	7
3.2 系统模块	7
4 硬件电路设计	8
4.1 电源	8
4.2 红外收发模块	8
4.3 直流步进电机驱动模块	8
5 工作原理	9
5.1 控制直流电机转动	9
5.2 读入红外收发模块信号	10
5.3 行驶	11
5.3.1 红外线检测状态	11
5.3.2 直行	11
5.3.3 向左转	12

5.3.4	向右转	12
5.3.5	停止	12
5.4	程序流	13
6	测试	14
6.1	调试和装配	14
6.2	软件修复	14
6.3	测试总结	14
7	困难及解决方案	16

1 任务安排

表 1.1 人员任务安排

任务安排	人员
核心代码	孙思进、肖茗怡
图片视频	肖茗怡
代码优化	涂梹
文档编写	杨怡林、涂梹

2 相关硬件说明

2.1 Arduino 单片机

2.1.1 Arduino Uno 介绍

我们小组的 Arduino 的版本使用的是 Uno, Arduino Uno 是 ArduinoUSB 接口系列的最新版本, 作为 Arduino 平台的参考标准模板。UNO 的处理器核心是 ATmega328, 同时具有 14 路数字输入/输出 (其中 6 路可作为 PWM 输出), 6 路模拟输入, 一个 16MHz 晶体振荡器, 一个 USB 口, 一个电源插座, 一个 ICSPheader 和一个复位按钮。

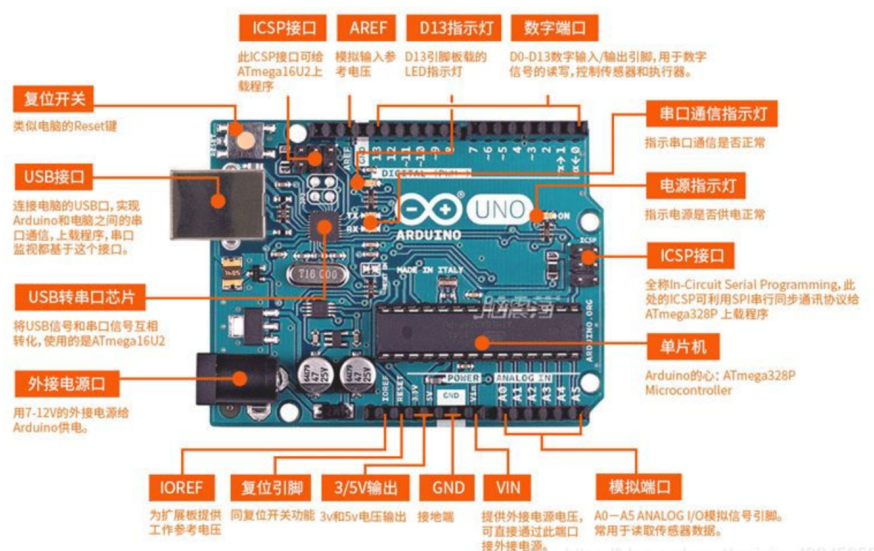
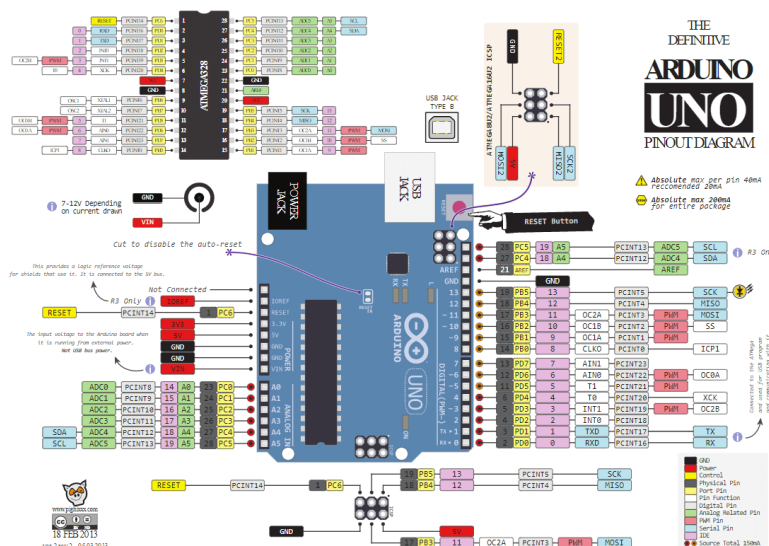


图 2.1 Arduino Uno 开发板

2.1.2 引脚介绍

- Power 引脚：开发板可提供 3.3V 和 5V 电压输出。
- Vin 引脚：可用于从外部电源为开发板供电。
- Analog In 引脚：模拟输入引脚，开发板可读取外部模拟信号，A0 A5 为模拟输入引脚。
- Digital 引脚：UNO R3 拥有 14 个数字 I/O 引脚，其中 6 个可用于 PWM(脉宽调制) 输出。

- TX 和 RX 引脚：标有 TX(发送) 和 RX(接收) 的两个引脚用于串口通讯。其中标有 TX 和 RX 的 LED 灯连接相应引脚，在串口通讯时会以不同速度闪烁。
- 13 引脚：开发板标记第 13 引脚，连接板载 LED 灯，可通过控制 13 引脚来控制 LED 灯亮灭。一般拿到开发板上电板载灯都会闪烁，可辅助检测开发板是否正常。
- GND：在该项目中 GND 即为电源的负极，同时要保证 Arduino 开发板，驱动模块等所有模块的 GND 连在一起才可以正常工作。



2.1.4 供电方式介绍

Arduino Uno 开发板可以使用三种方式供电，直流电源插孔、VIN 引脚以及 USB 电缆。在这里我们使用的是直流电源插孔的方式对其进行供电。

电源插孔通常连接到一个适配器，开发板的供电范围可以是 5-20V，但制造商建议将其保持在 7-12V 之间。高于 12V 时，稳压芯片可能会过热，低于 7V 可能会供电不足。这里我们使用老师提供的电池即可满足要求。

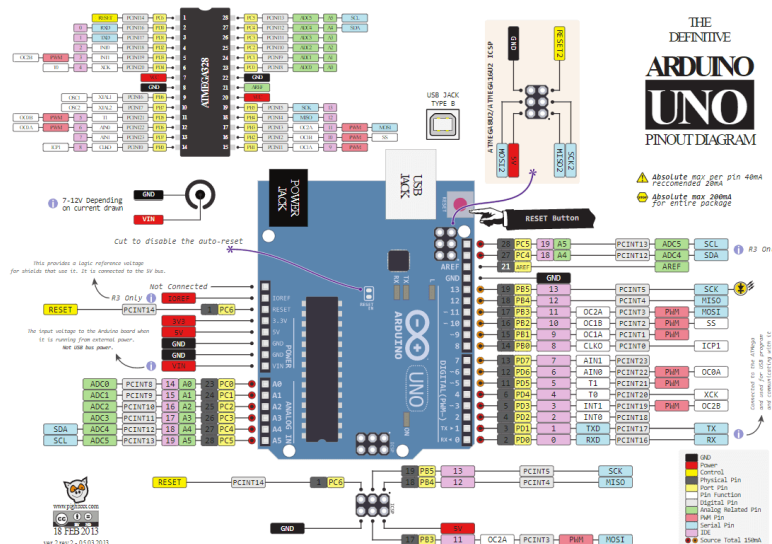


图 2.4 Arduino Uno 引脚分配 - 电源

2.2 红外收发模块

我们小组使用的是知道老师提供的 TCRT5000 红外传感器，TCRT5000 传感器的红外发射二极管不断发射红外线，当发射出的红外线没有被反射回来或被反射回来但强度不够大时，红外接收管一直处于关断状态。

此时模块的输出端为低电平，指示二极管一直处于熄灭状态，被检测物体出现在检测范围内时，红外线被反射回来且强度足够大，红外接收管饱和，此时模块的输出端为高电平，指示二极管被点亮。

2.2.1 模数转换

Arduino Uno 有 6 个模拟引脚，它们作为 ADC（模数转换器）使用。这些引脚用作模拟输入，但也可用作数字输入或数字输出。

ADC 表示模拟到数字转换器。ADC 是用于将模拟信号转换为数字信号电子电路。模拟信号的这种数字表示允许处理器（其是数字设备）测量模拟信号并在其操作中使用它。Arduino 引脚 A0-A5 能够读取模拟电压。

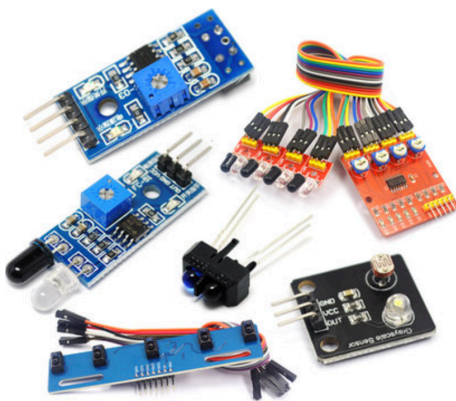


图 2.5 Arduino Uno 引脚分配 - 模拟输入

2.2.2 数字端口介绍

Arduino Uno 的引脚 0-13 用作数字输入/输出引脚。其中，引脚 13 连接到板载的 LED 指示灯；引脚 3、5、6、9、10、11 具有 PWM 功能。

脉宽调制（PWM）是一种调制技术，用于将消息编码为脉冲信号。PWM 由两个关键部分组成：频率和占空比。PWM 频率决定了完成单个周期（周期）所需的时间以及信号从高到低的波动速度。占空比决定信号在总时间段内保持高电平的时间。占空比以百分比表示。在 Arduino 中，支持 PWM 的引脚产生约 500Hz 的恒定频率，而占空比根据用户设置的参数而变化。PWM 信号用于直流电机的速度控制，调光 LED 等。

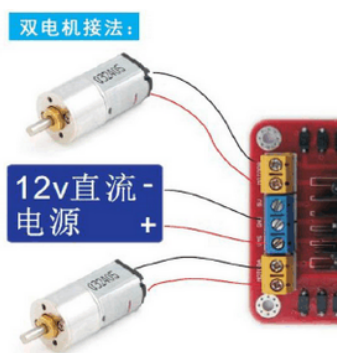


图 2.6 电机驱动

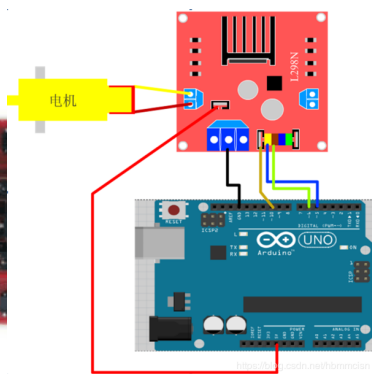


图 2.7 电机相连示意

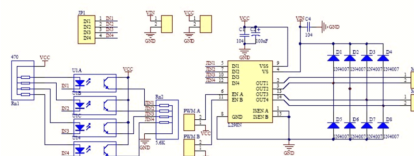


图 2.8 L298N 的原理图

2.3 L298N 介绍

L298N 是一款接受高电压的电机驱动器，直流电机和步进电机都可以驱动，其芯片工作电压：DC 4.5 5.5V，电机驱动电源电压 DC 5-35V。一片驱动芯片可同时

控制两个直流减速电机做不同动作，在 6V 到 46V 的电压范围内，提供 2 安培的电流，并且具有过热自断和反馈检测功能。电源输入输出正常时有 LED 灯指示。L298N 可对电机进行直接控制，通过主控芯片的 I/O 输入对其控制电平进行设定，就可为电机进行正转反转驱动，操作简单、稳定性好，可以满足直流电机的大电流驱动条件。

3 小车结构与系统模块设计

3.1 小车结构

如下图3.1所示，小车前方有三个红外收发模块用于检测路径；主体左部分上放有 L298N 直流步进电机驱动模块，用于调节直流电机的转速；主体右部分放有 Arduino Uno 作为主控。有 9V 锂电池和 4 节 1.5V 干电池作为电源。左右两边分别有两个直流电机分别调节左轮与右轮的转速。

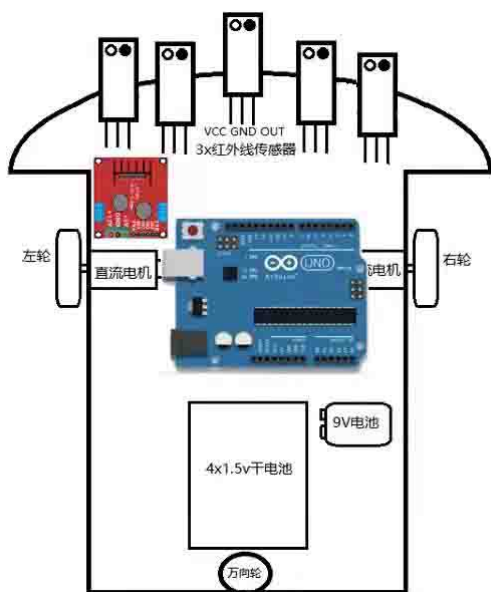


图 3.1 小车整体结构

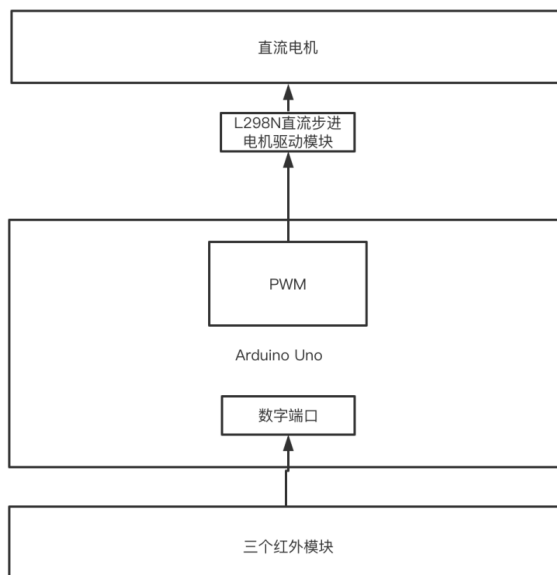


图 3.2 系统模块

3.2 系统模块

如图3.2所示，Arduino Uno 通过数字端口接收三个红外收发模块的数据，通过分析传来的数据，输出 PWM 数字信号到直流步进电机驱动模块，来控制左右电机的转速。

4 硬件电路设计

4.1 电源

如图4.1所示，9v 充电电池接到直流步进电机的 9V 输入口，为直流电机供电，同时接入 Arduino Uno 的 DC 插头，为开发版供电；5 个红外收发模块的 VCC 输入口与电机的 +5V 输入口、ENA 输入口、ENB 输入口，接入到 Arduino Uno 的 +5V 输出口；5 个红外收发模块的 GND 输入口与电机的 GND 输入口接入到 Arduino Uno 的 GND 输出口上；两个直流电机的正负极分别接入到电机的马达 A 输出口与马达 B 输出口上。

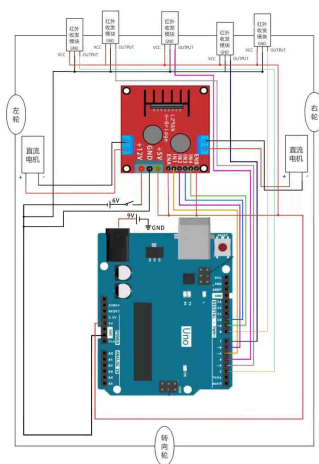


图 4.1 电路原理图

4.2 红外收发模块

如图4.1所示，将五个红外收发模块的 OUTPUT 输出口接入到 Arduino Uno 数字输出口，向 Arduino Uno 实时传送检测到的数据。

4.3 直流步进电机驱动模块

如图4.1所示，将电机的输入口分别接入到 Arduino Uno 的带有 PWM 功能的数字输出口上。Arduino Uno 通过分析红外收发模块传来的数据，通过上述端口控制直流电机驱动模块，直流电机驱动模块再控制直流马达的转动。

5 工作原理

5.1 控制直流电机转动

Arduino 开发版的 5、6、10、9 号带有 PWM 功能的数字输出端口传递控制信号到 L298N 控制电机转动。

```
1  /*电机引脚定义*/
2  #define motor_LIN1 11
3  #define motor_LIN2 10
4  #define motor_RIN1 3
5  #define motor_RIN2 9
6
7  /*引脚初始化*/
8  void motor_pin() {
9      pinMode(motor_LIN1,OUTPUT);
10     pinMode(motor_LIN2,OUTPUT);
11     pinMode(motor_RIN1,OUTPUT);
12     pinMode(motor_RIN2,OUTPUT);
13 }
14 /*左大转弯*/
15 void turn_1_left() {
16     analogWrite(motor_LIN1,0);
17     analogWrite(motor_LIN2,0);
18     analogWrite(motor_RIN1,100);
19     analogWrite(motor_RIN2,0);
20 }
21 /*左小转弯*/
22 void turn_2_left() {
23     analogWrite(motor_LIN1,0);
24     analogWrite(motor_LIN2,0);
25     analogWrite(motor_RIN1,60);
26     analogWrite(motor_RIN2,0);
27 }
```

```

28 void turn_left90 () {
29     analogWrite(motor_LIN1,0);
30     analogWrite(motor_LIN2,100);
31     analogWrite(motor_RIN1,100);
32     analogWrite(motor_RIN2,0);
33 }
34
35 /*右大转弯*/
36 void turn_1_right () {
37     analogWrite(motor_LIN1,100);
38     analogWrite(motor_LIN2,0);
39     analogWrite(motor_RIN1,0);
40     analogWrite(motor_RIN2,0);
41 }
42
43 /*右小转弯*/
44 void turn_2_right () {
45     analogWrite(motor_LIN1,60);
46     analogWrite(motor_LIN2,0);
47     analogWrite(motor_RIN1,0);
48     analogWrite(motor_RIN2,0);
49 }
50 void turn_right90 () {
51     analogWrite(motor_LIN1,100);
52     analogWrite(motor_LIN2,0);
53     analogWrite(motor_RIN1,0);
54     analogWrite(motor_RIN2,100 );
55 }

```

motor 函数可直接控制左右两个直流电机的转动，如 motor(-240, 120) 代表左轮反转，右轮正转，240 经 PWM 后可调节左轮反转的速度，120 经 PWM 后可交接右轮正转速度。

5.2 读入红外收发模块信号

为了保证小车判断的稳定性，我们使用五个红外模块，以下是代码定义。

```

1 /*寻迹模块从左到右*/

```

```

2 #define sensor1 4
3 #define sensor2 5
4 #define sensor3 6
5 #define sensor4 7
6 #define sensor5 8

```

5.3 行驶

5.3.1 红外线检测状态

其中“1”表示检测到黑线，“0”代表没有检测到黑线。

表 5.1 小车状态示意图

Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	小车状态
0	0	0	1	1	左转 90 度
1	1	0	0	0	右转 90 度
0	1	1	1	1	左大转弯
0	0	1	1	1	左大转弯补充状态
1	0	1	1	1	左小转弯
1	0	0	1	1	直行补充状态
1	1	0	1	1	直行
1	1	1	0	1	右小转弯
1	1	0	1	1	直行补充状态
1	1	1	1	0	右大转弯
1	1	1	0	0	右大转弯补充状态
0	0	0	0	0	停止

5.3.2 直行

如表5.1所示，当且仅当红外模块 3 检测到路径时小车直行。

```

1 /* 直行 */
2 void forward() {
3     analogWrite(motor_LIN1, 100);
4     analogWrite(motor_LIN2, 0);

```

```

5   analogWrite(motor_RIN1,100);
6   analogWrite(motor_RIN2,0);
7   }

```

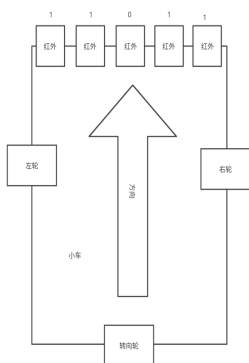


图 5.1 直行示意图

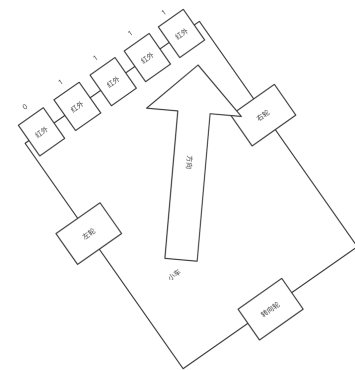


图 5.2 向左转示意图

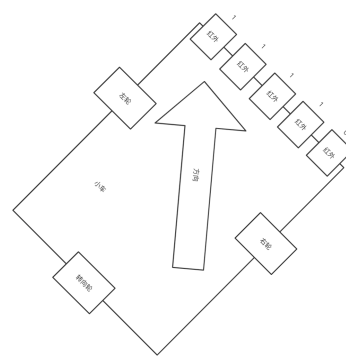


图 5.3 向右转示意图

5.3.3 向左转

如图5.2所示，当只有红外模块 1 检测到路径时，小车向左调整直到红外模块 2 检测到路径。

```

1   else if ((data[0] == 0) && (data[1] == 1) && (data[2]
2       == 1) && (data[3] == 1) && (data[4] == 1))
       error = -2;

```

5.3.4 向右转

如图5.3所示，当红外模块 5 检测到路径时，小车向右调整直到红外模块 3 检测到路径。

```

1   else if ((data[0] == 1) && (data[1] == 1) && (data[2]
2       == 1) && (data[3] == 1) && (data[4] == 0))
       error = 2;

```

5.3.5 停止

当只有红外模块 1、2、3、4、5，同时检测到无路径时，小车会保持原状，如果同时检测到有路径时，会停止当前状态，表示到达终点。


```

1  else if ((data[0] == 0) && (data[1] == 0) && (data[2]
2    == 0) && (data[3] == 0) && (data[4] == 0))
    error = 5;

```

5.4 程序流

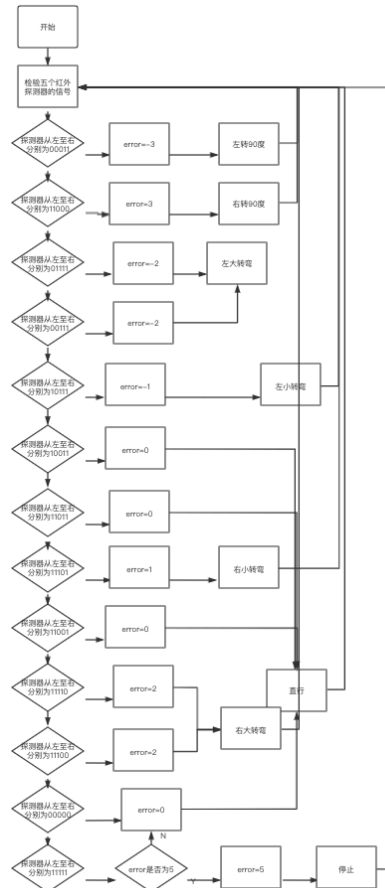


图 5.4 电路原理图

6 测试

6.1 调试和装配

- 1) 首先尝试用 PIC12F683 来控制小车的前进、后退、左右转向。结果试验成功，小车能正常的跑动起来。成功检测小车的机械性能可以达到我们预期的目的。
- 2) 制作主板
- 3) 主板通电前检查：电路安装完毕，我们首先直观检查电路各部分生产线是否正确，检查电源、地线、信号线、元器件引脚之间有无短路，器件有无接错。结果发现 L298N 芯片的 5 和 12 管脚没有接地，查看发现原来是原理图没注意改正才导致的。
- 4) 通电检查：给电机通电，观察电机是否工作正常。电机正常工作时，后驱工作电流为 320 mA，电压为 5.4V；前驱电机工作电流为 180mA，电压为 5.41V。给主板通电，观察电路各部分器件有无异常现象。
- 5) 主板安装调试（主控芯片用 PIC16F886），在调试的过程中我们发现了原理图中有一个小小的错误。这个错误导致小车的稳压芯片过热。一起讨论之后决定，修改原理图，调换小车驱动芯片的位置。改动之后，小车电源稳压芯片过热现象消失。小车也能实现了基本的功能。（由于我们没能在制板之前发现这个问题，导致了主板在修整后变得不太美观了。）
- 6) 制作并调试小车循迹板，传感器采用反射式红外传感器 ST178，当 $V_{cc}=5V$ 时，工作电流为 17mA, $VR1=3.83V$, $VD=1.17V$, $RD=68.8$ 欧。

6.2 软件修复

在调试的过程中却发现小车不循迹，经过了很久的努力，终于发现了问题的所在，原来是代码控制逻辑写错了导致（错将判断条件为真当成为假使用的来用）。改正后调试，终于实现了循迹功能。

6.3 测试总结

测试结果表明：本组智能小车能很好的完成了基本功能和循迹功能，跑道是由黑色胶布在白色地面上拉线完成，小车可以从 O 型跑道的任何段为起点，跑完全程。本组智能小车目前只能完成在没有任何交叉路口的跑道上，要想跑在有“十”

字路口的跑道上，还须要对循迹程序进行补充和修改。我们保留了各种硬件接口和软件子程序接口，方便以后的扩展和进一步的开发。

7 困难及解决方案

- 1) 小车走得过慢或者有的弯不能贴合指定路线。

烧入电机控制的小程序有的指令用的不正确，导致电路功能不能完全实现，另外软件程序中的延时有的过长、有的过短。

- 2) 红外无法检测旁边的障碍物。

将小车舵机上的红外设置为八字形。首先八字形具有一定的美观效果，再者，可以避免设置成一字型撞到障碍物，最后就是可以使小车在行走的时候同时也能检测旁边的障碍物。

- 3) 在十字路口或者岔路口前进的时候小车会失去方向。

通过后续代码的优化，在十字路口或者岔路口时，小车会首先直走。