



西安财经大学
XI'AN UNIVERSITY OF FINANCE AND ECONOMICS

嵌入式微处理器分析与应用大作业

题目：基于 STC89C52STC 单片机的简易智能电动小车的设计

学生姓名：张探探

学 号：1931053726

专 业：计算机科学与技术

班 级：1902

指导教师：魏晋雁

完成日期：2022 年 6 月 17 日

《嵌入式微处理器分析与应用大作业》成绩评定表

班级：计本 1902 姓名：张探探 学号：1931053726 阅卷教师签名：

设计任务	评分项	分值	子项评分标准及分值	得分
设计实现部分	总体设计	20	任务分析明确（5）	
			方案设计合理有新意（10）	
			软件和硬件功能划分合理（5）	
	硬件设计	20	片内器件分配正确、合理（5）	
			电路原理图设计正确（10）	
			电路图布线整齐、合理（5）	
	软件设计	20	算法和数据结构设计正确、合理（5）	
			流程图设计正确、简明（5）	
			程序设计正确、有新意（10）	
	设计实现	20	调试顺序正确（5）	
			能熟练排除错误,回答问题正确(10)	
			调试后运行正确（5）	
设计报告部分	设计报告	20	书写规范、整齐（5）	
			内容详实具体（5）	
			图形绘制正确、完整、全面（5）	
			能正确客观评价分析设计结果（5）	
总成绩				

目录
目录

一、设计要求及器件选择.....	4
1.1 设计要求.....	4
1.2 主要器件选择.....	4
1.2.1 基础器件.....	4
1.2.2 直流电机.....	4
1.2.3 L298N.....	5
1.2.4 避障碍传感器模块.....	6
1.2.5 TCRT5000.....	6
1.2.6 51 单片机最小系统版.....	7
2 系统总体方案设计.....	8
2.1 总体设计思路.....	8
1. 要求一:.....	8
2. 要求二:.....	8
3. 要求三:.....	8
4. 要求四:.....	8
2.2 方案选择.....	9
2.2.1 方案一: 采用 E18-D80 作为避障模块.....	9
2.2.2 方案二: 采用步进电机作为小车的驱动电机.....	9
2.2.3 方案三: 采用超声波测距模块.....	9
3 系统硬件设计.....	10
3.1 电源.....	10
3.2 时钟.....	10
3.3 复位.....	11
3.4 蜂鸣器.....	11
3.5 TCRT5000.....	12
3.6 红外避障碍传感器模块.....	12
4 系统软件设计.....	14
4.1 流程图.....	14
4.2 程序宏定义.....	14
4.3 全局变量.....	15
4.4 前进模块.....	15
4.5 报警和警示灯.....	16
4.6 中断.....	16
5 系统调试及仿真.....	18
5.1 软件调试.....	18
5.1.1 蜂鸣器调试.....	18
5.1.2 Logic Input.....	18
5.1.3 电机转速的调节.....	19
5.2 硬件调试.....	20
5.2.1 杜邦线的连接.....	20
5.2.2 电机的固定以及测试.....	21
5.2.3 避障模块调试.....	21

5.2.4 TCRT5000 测试.....	22
5.2.5 系统总体运行图.....	22
5.3 系统总体性能分析.....	23
6 设计总结.....	24
参考文献.....	25
附录.....	26
附录 1 源码.....	26
附录 2 最小系统版原理图.....	29

一、设计要求及器件选择

1.1 设计要求

用 STC89C52 单片机作为核心控制元件，LED 数码管显示，设计一个简易智能电动小车，具体要求如下：

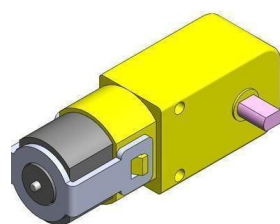
1. 上电后，按下启动键，小车从自定义的椭圆型轨道起点出发，能够自己识别轨道黑色道路，平稳行驶。
2. 到达转弯的地方，小车自行判断方向，灵活转弯。
3. 遇到轨道中的障碍，可以实行避让或停车，并闪烁报警灯，同时蜂鸣器响，绕开障碍后继续前进。
4. 到达终点进入车库行驶，不得碰到车库内壁，LED 数码管显示总用时。

1.2 主要器件选择

1.2.1 基础器件

大小适合的小车主板，杜邦线若干，热容胶若干，螺丝螺母若干，电池组，C52 单片机主板。

1.2.2 直流电机

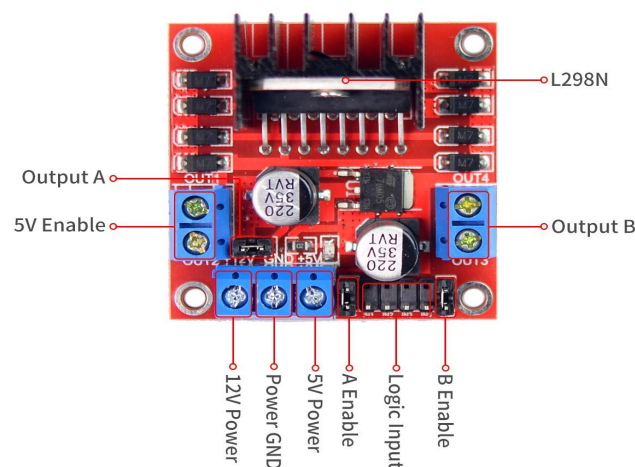


直流电机将直流电能用来旋转电机，可以实现直流电能与机械能相互转换的电机，直流电机是两线连续旋转电机，两线分别是电源和地线。接通电源后，直流电机将开始旋转，直到断开电源。电机速度可以通过使用 PWM 来控制，（在本项目中，采取对模拟信号的近似，使用数字信号来模拟）。而直流电机对控制电机的顺时针旋转和逆时针旋转极为方便。

1.2.3 L298N

L298N 有以下优点：

1. 发热量很低
2. 抗干扰能力强
3. 额定工作电压范围大
4. 输出电流大



以下对所接引脚进行介绍：

1. 电源引脚

- (1) 12V Power：电源输入口，支持的电压范围:5-35V。
- (2) Power GND：驱动地线接入口。
- (3) 5V Power：电源输出口，将输入的电压稳定转换为 5V，本项目中主要对单片机供电。

2. 使能端

- (1) A Enable：A 的使能端，主要由跳线帽控制，若扒开跳线帽则使能关闭。
- (2) B Enable：B 的使能端，主要由跳线帽控制，若扒开跳线帽则使能关闭。

3. 输入端（想要使得输入端有效，必须保证使能端有效）

Logic Input：分为四个引脚，分别控制电机顺时针和逆时针旋转。

(1) IN1 & IN2

- ① IN1 输入高电平，IN2 输入地电平，则左电机正转
- ② IN1 输入低电平，IN2 输入高电平，则左电机倒转

(2) IN3 & IN4

- ① IN3 输入高电平，IN4 输入地电平，则右电机正转
- ② IN3 输入低电平，IN4 输入高电平，则右电机倒转

4. 输出引脚

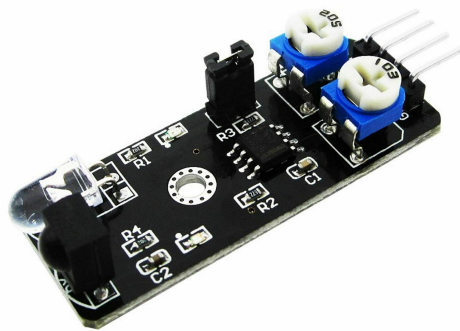
- (1) OutPut A: 连接左电机
- (2) OutPut B: 连接右电机

注：影响电机所转的方向因素很多，而直流电机正负接反也会导致旋转方向相反，需要读者自行尝试。

1.2.4 避障碍传感器模块

该模块的优点：

- 1. 距离远
- 2. 红外传感器
- 3. 光电开关



1.2.5 TCRT5000



产品介绍：

- 1. 检测反射距离：1mm ~ 25mm

2. 比较器输入，信号干净，波形好，驱动能力强，超过 15mA
3. 配电位器调节灵敏度
4. 工作电压 3.3V 5B
5. 输出形式：数字开关输出 0/1

注意事项：

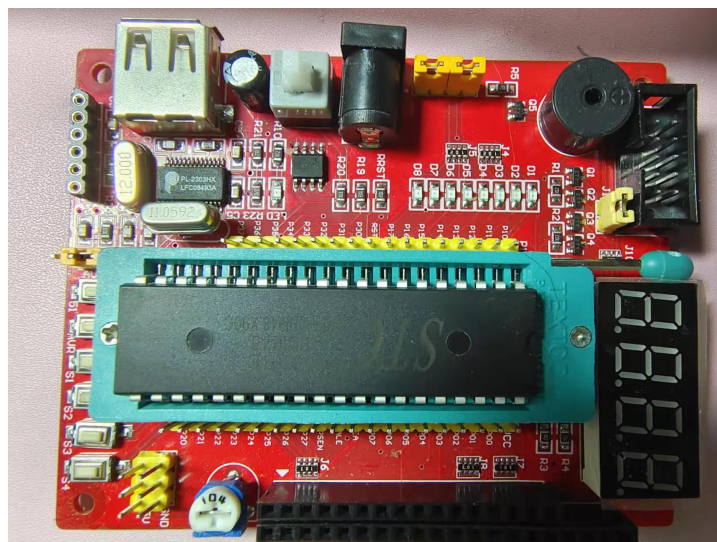
1. 使用中光电传感器的前端面与被检测的工作或物体表面必须保持平行，这样光电传感器的转换效率最高。
2. 光电传感器必须安装在没有强光直接照射的地方，因为强光中的红外光将影响接收管的正常工作。
3. 光电传感器长时间工作时红外接收管的最大工作电流不应超过 250uA。

引脚：

1. VCC 接电源正极
2. GND 接电源负极
3. D0 TTL 开关信号输入

注：碰到黑色输出高电平，注意是存在有效范围的，下限不为 0

1.2.6 51 单片机最小系统版



而选择 C51 作为小车的主控版，一方面由于 C51 价格低廉，且容易上手学习，另外一方面，以 C51 的运算速度，完全足够整个小车的使用。

2 系统总体方案设计

2.1 总体设计思路

本设计选用 C51 作为小车的主控版，借助 L298N 对单片机和部分模块供电。

1. 要求一：

小车通过 3 个 TCRT5000 来探测轨道，分为置于左中右三个方向，正常行驶情况下，处于中间的 TCRT5000 持续扫描到黑线并持续返回高电平，处于左侧和右侧的 TCRT5000 扫描黑线的两侧，持续返回低点平。

2. 要求二：

当中间的 TCRT5000 返回低电平，说明已经偏移轨道，那么我们扫描左右两侧谁扫描到黑线，我们就向相反的方向旋转，使之返回到正确的道路中当小车需要左转时，我们控制 Logic Input 使得左轮后转，右轮前转使得小车快速掉头，同理，当小车需要右转的时候，我们控制 Logic Input 使得右轮后转，左轮前转以达到相同的目的。

3. 要求三：

当处于高位的避障碍传感器模块检测到前方的障碍物的时候，返回低点平，单片机收到信号后，控制小车停车，并打开蜂鸣器和报警灯，并程序轮询的方式持续检测避障碍传感器模块返回的信号，若收到高电平，表明前方以无障碍物，关闭蜂鸣器和报警灯，继续前行。

4. 要求四：

当小车车头的三个 TCRT5000 同时检测不到黑线时，认为到达车库，停车并显示 LED 数码管的值即可。

2.2 方案选择

2.2.1 方案一：采用 E18-D80 作为避障模块

E18-D80 一种非常便宜、易于组装、易于使用的红外传感器，检测距离远，受可见光干扰小。调制 IR 信号的实现使传感器免受灯泡的正常光线或太阳光引起的干扰。该传感器具有螺丝刀调节功能，可设置适当的检测距离，使其适用于许多应用，然后在检测到该范围内的某些物体时提供数字输出。该传感器不测量距离值。可用于防撞机器人和机器自动化。传感器提供非接触式检测。

E18-D80 的性能比 TCRT5000 更适合用在这个地方，但由于成本过高，故最终选择了 TCRT5000 作为避障的模块。

2.2.2 方案二：采用步进电机作为小车的驱动电机

而与直流电机不同的是，步进电机是将电脉冲信号转变为角位移或线位移的开环控制元件，在额定功率下，转速取决于脉冲信号的频率和脉冲数。而步进电机适用于精确度很高的场合，比如对旋转的角度或旋转周期有严格的要求，比较适合使用步进电机。而本项目并不需要如此高的精确度，对设计要求来说，本项目采取的解决办法是通过小车前端的检测模块传回的数据来判断该转向的方向，故本项目采用直流电机作为小车的驱动电机。

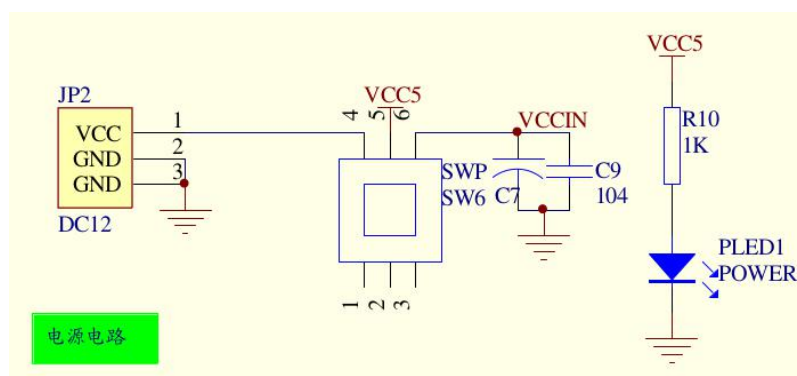
2.2.3 方案三：采用超声波测距模块

超声波测距模块控制端口发射一个 10US 以上的高电平，就可以在接收口等待高电平输出，一有输出就可以开定时器计时，当此口变成低电平就可以读定时器的值，此时接位此次的测距的时间，方可算出距离。

但在设计要求中，并未要求显示距离，故我们采取成本更低的 TCRT5000 作为避障模块。

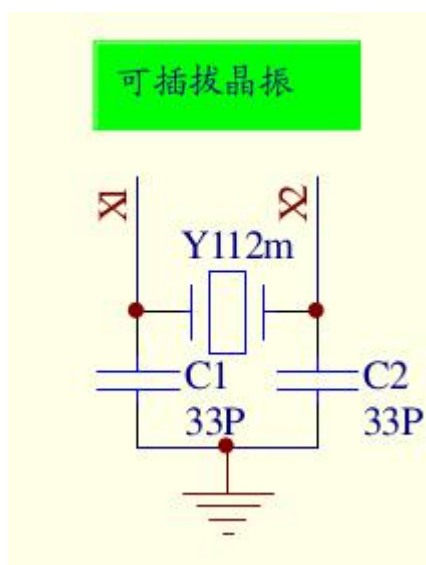
3 系统硬件设计

3.1 电源



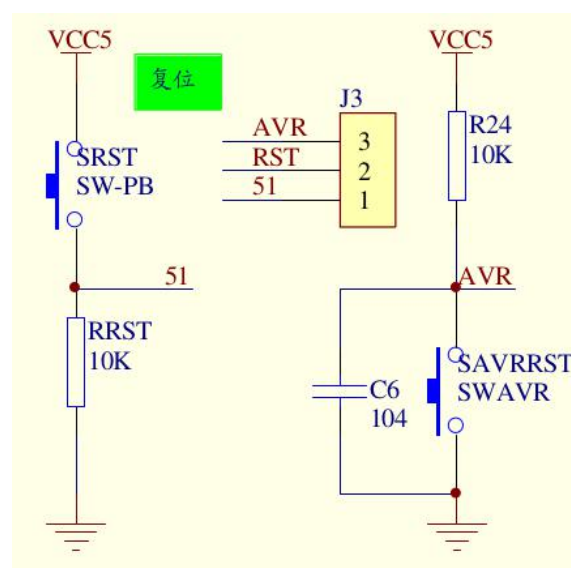
电源可以通过 DC 插座和 USB 插座接入，接入后连接到开关，开关可以起到互锁的功能，同时只能有一路电源进入开发板，经过开关的电源输出经 LED 指示电源状态。

3.2 时钟



晶振、电容 C1、C2 共同构成了电容振荡器，而电容 C1、C2 的容量影响着振荡频率以及晶振的频率，主要决定因素是晶振的振荡频率。范围在 0MHz ~ 33MHz 之间，而电容 C1、C2 在 5pF ~ 30pF 之间取值 [3]。本设计中采取 12MHz 作为系统的外部晶振。

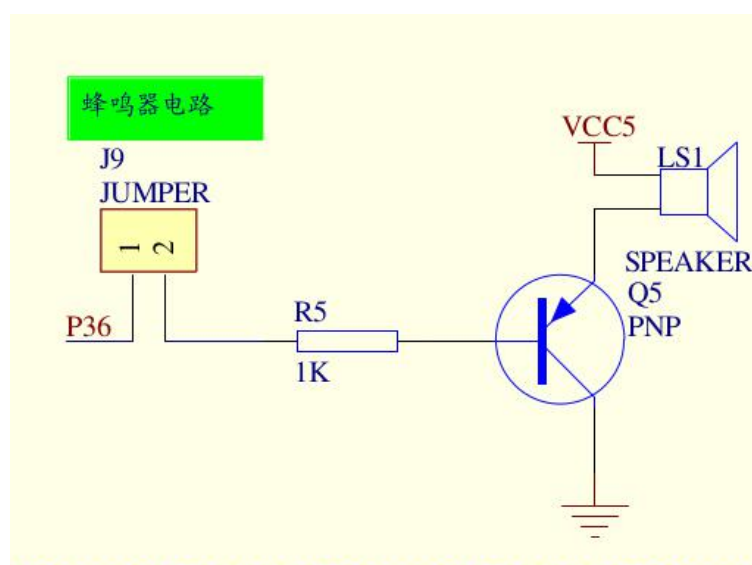
3.3 复位



单片机复位电路可以说是一个基本元件，确定初始状态的元件，在单片机刚上电，断电后以及发生故障都需要进行复位，在复位过程中，电容处于短路状态，释放所有电能，实现复位。

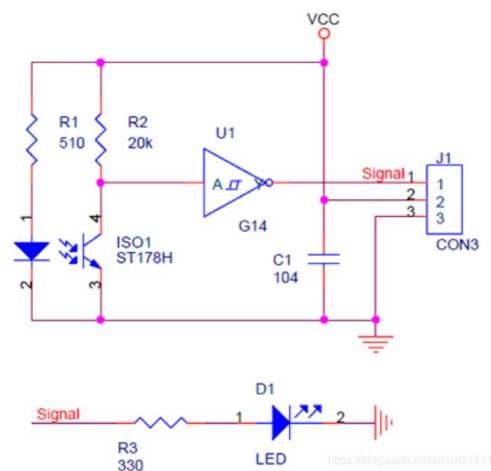
这个复位电路在最基础的复位电路上增加了按键复位，也就是说在运行过程中，可以通过手动按压复位键来实现复位，而不必断电重启，极大的方便了测试以及使用。并且还可以避免死锁，实现可靠运行。

3.4 蜂鸣器



蜂鸣器分为有源和无源两种，由于 C51 内嵌了蜂鸣器，所以就使用了 C51 内嵌的无源蜂鸣器，而无源蜂鸣器的使用方式，我们需要波来实现，例如:P3_6 是蜂鸣器的控制端口，我们需要不断提供 01 的数字信号来模拟波，以发出持续的声音，但就本项目来说，有源蜂鸣器更适合，因为本项目仅仅需要报警，而不必发出复杂声音，但由于 C51 内嵌了无源蜂鸣器，我们只要固定模拟波，那么实际上这里的无源蜂鸣器等同于有源蜂鸣器。

3.5 TCRT5000

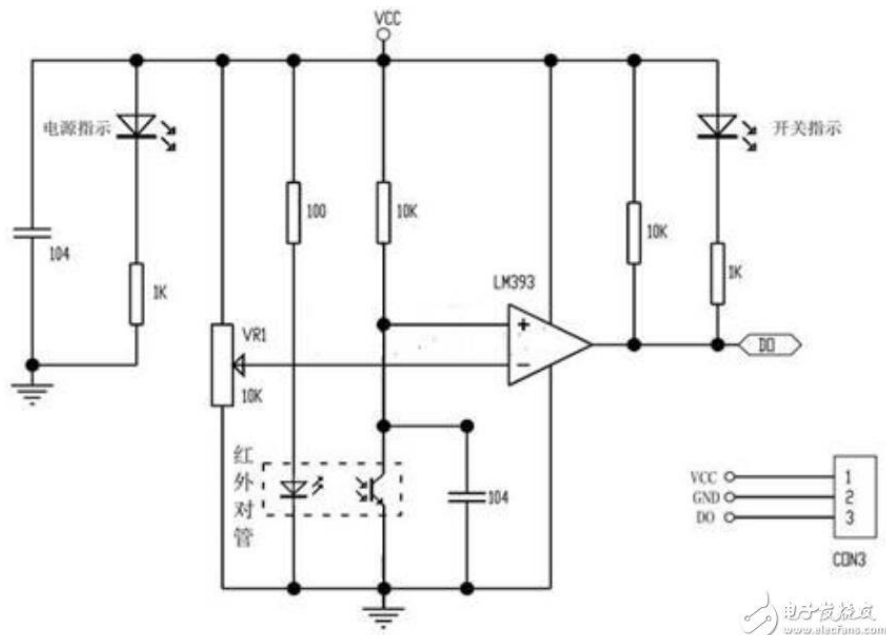


传感器的红外发射二极管不断发射红外线，当发射出的红外线没有被反射回来或被反射回来但强度不够大时，光敏三极管一直处于关断状态，此时模块的输出端为低电平，指示二极管一直处于熄灭状态；被检测物体出现在检测范围内，红外线被反射回来且强度足够大。光敏三极管饱和，此时模块的输出端为高电平，即二极管被点亮。

3.6 红外避障碍传感器模块

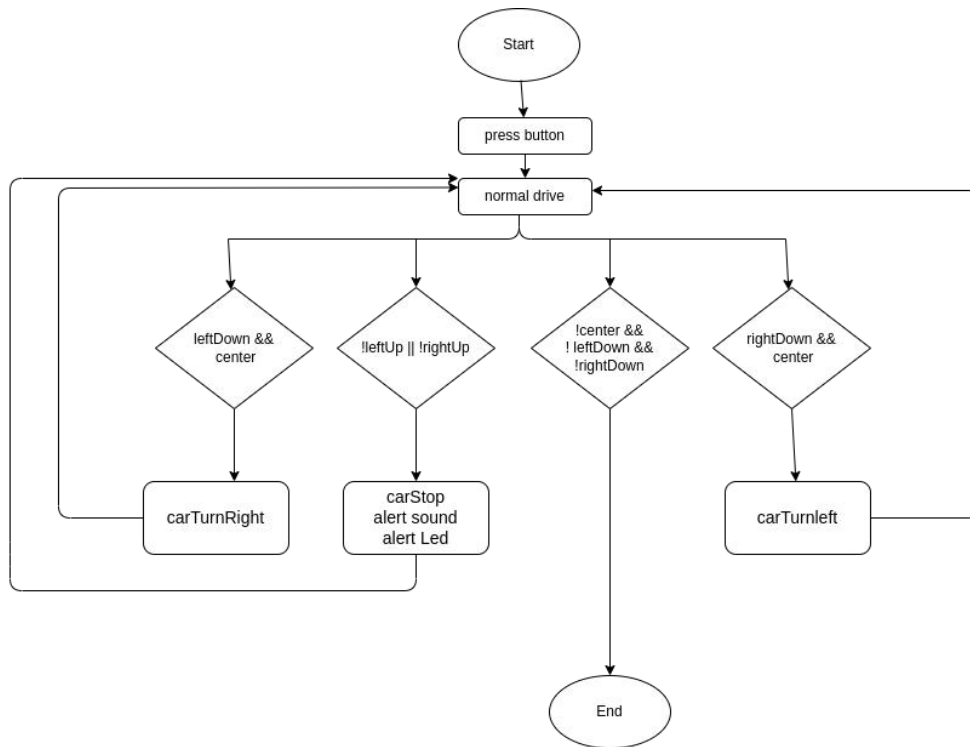
1. 感应到物体输出 0，未感应到输出 1。可以直接与 3V 和 5V 单片机 I/O 口相连
2. 感应距离 2 40cm，距离远，抗干扰能力强
3. 由于是光电元件，是检测光线反射率，所以不同的物体或许得到的距离不同，物体颜色越深越接近黑色，距离越短。
4. 传感器 3V-6V 供电，范围宽，适合 3V 到 5V 的单片机

5. 使能端，EN 等于 1 的时候传感器不工作，等于 0 的时候工作，传感器上跳线帽，保证 EN 持续接地，要使用 EN 的话，需要拔掉跳线帽。



4 系统软件设计

4.1 流程图



4.2 程序宏定义

```
#define leftWheelFront P2_4=1;P2_5=0;
#define rightWheelFront P2_6=1;P2_7=0;
#define carStop P2_4=1;P2_5=1;P2_6=1;P2_7=1;
#define carTurnRight P2_4=0;P2_5=1;P2_6=1;P2_7=0;
#define carTurnLeft P2_4=1;P2_5=0;P2_6=0;P2_7=1;
#define leftUp P1_0
#define rightUp P1_1
#define leftDown P1_2
#define rightDown P1_3
#define center P1_4
#define sound P2_0
#define alertLed P1_0
```

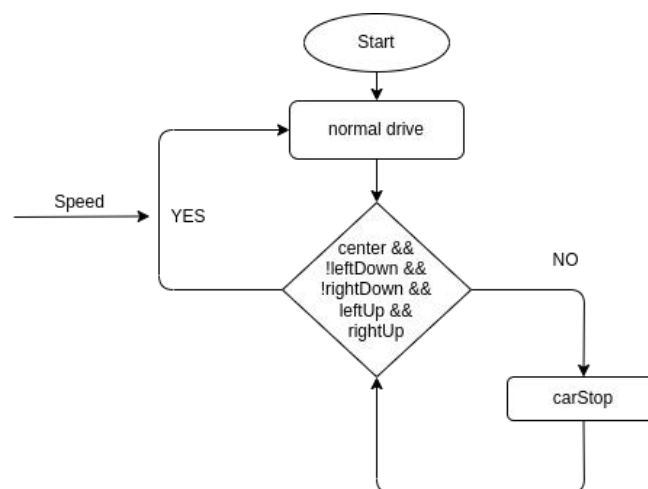
这些宏定义可以使得代码阅读更加方便，也同时给出了物理接口的连线方式。

4.3 全局变量

```
int count = 0;
int time = 0;
char i = 0;
char table[]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,
0xf8,0x80,0x90};
char End = 0;
```

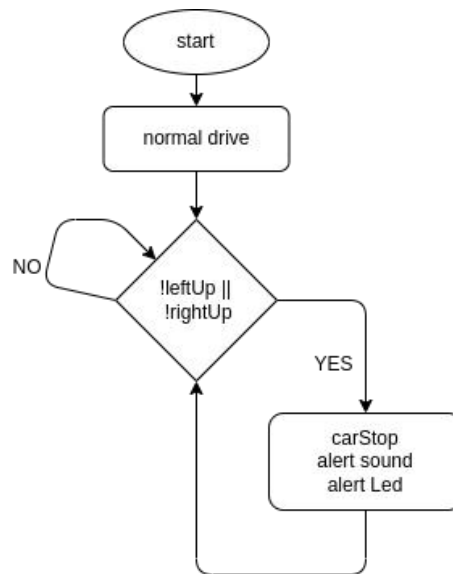
1. count 是用来计时器所用的计时，每进一次中断，就加一，直到加到一秒后清零。
2. time 是当前小车所跑的秒数。
3. i 是中断中为了减少运行中断函数的时常，所加的 switch-case 中的筛选变量。
4. End 是结束标志，表明到达车库。

4.4 前进模块



```
// car go front
void go(char speed){
for(char a = 0; a < speed; ++a){
carStop
}
leftWheelFront
rightWheelFront
}speed 主要用在减速，speed 越大，小车跑的越慢。
```


4.5 报警和警示灯



```
// alert sound
void sound() {
    sound = 0;
    for(char a= 0; a <= 10; ++a);
    sound = 1;
}

// alert Led
void AlertLed() {
    alertLed = 0;
}
```

C51 内嵌的无源蜂鸣器的控制口为 P3_6，即为 sound，通过延迟 10 个单位的 for 循环来模拟波，使得无源蜂鸣器开始振动。而 alertLed 是 P1_0 的 Led 的控制端。

4.6 中断

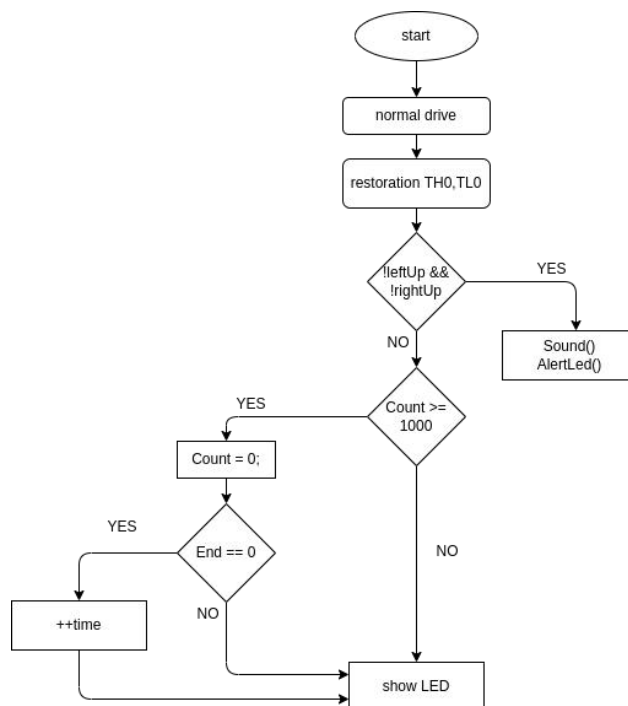
```
void interrupt_() __interrupt 1{
    TH0 = 0xFC;
    TL0 = 0x67;
    ++count;
```

```

if(leftUp == 0 || rightUp == 0){
Sound();
AlertLed();
}else
alertLed = 1;
if (count >= 1000){
count = 0;
if(!End)
time++;
}
switch (i){
case 0: P2_0 = 1; P0 = table[time % 10]; P2_1 = 0; ++i;
break;
case 1: P2_1 = 1; P0 = table[time % 100 / 10]; P2_0 = 0; --i;
break;
}
}

```

中断主要是来程序轮询查看是否到达车库，以及遇到障碍物的停车，报警以及警示灯。



5 系统调试及仿真

5.1 软件调试

5.1.1 蜂鸣器调试

```
// alert sound
void sound() {
    sound = 0;
    for(char a= 0; a <= 10; ++a);
    sound = 1;
}
```

蜂鸣器由于采取的是无源蜂鸣器,所以需要不断调整输入 01 来实现数字信号对 PWM 的模拟,以纯程序控制开关的方式来操作蜂鸣器,可以调整延时的时间 即调整。

```
for(char a= 0; a <= 10; ++a);
```

可以将 10 修改为其他的值,使得蜂鸣器的声调发生变化。

5.1.2 Logic Input

对电机的控制借助 L298N 的逻辑输入口 Logic Input 来实现控制,假设在当前的物理连线不变的情况下, $IN1 = 1, IN2 = 0$ 为前转,那么我们就可以肯定的是,我们选择 $IN1 = 0, IN2 = 1$ 必定为后转,而其他的两种状态,

$$IN1 = 0, IN2 = 0$$
$$IN1 = 1, IN2 = 1$$

下,小车的电机是不会转动的,同理, $IN3$ 和 $IN4$ 也是如此。

而本项目最终调试结果为:

```
#define leftWheelFront P2_4=1;P2_5=0;
#define rightWheelFront P2_6=1;P2_7=0;
#define carStop P2_4=1;P2_5=1;P2_6=1;P2_7=1
#define carTurnRight P2_4=0;P2_5=1;P2_6=1;P2_7=0;
#define carTurnLeft P2_4=1;P2_5=0;P2_6=0;P2_7=1;
```

5.1.3 电机转速的调节

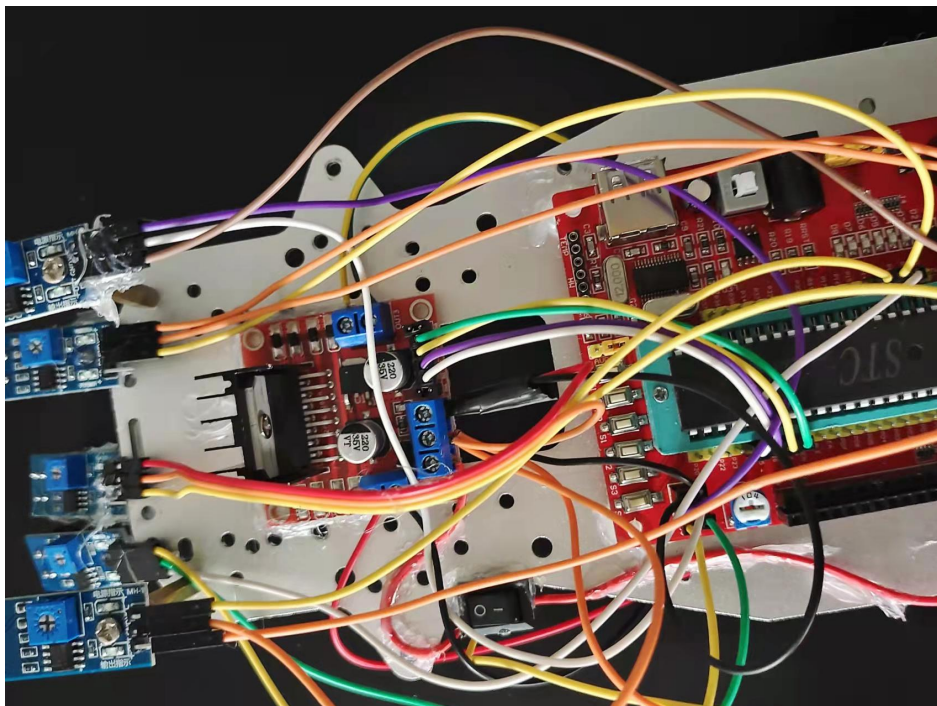
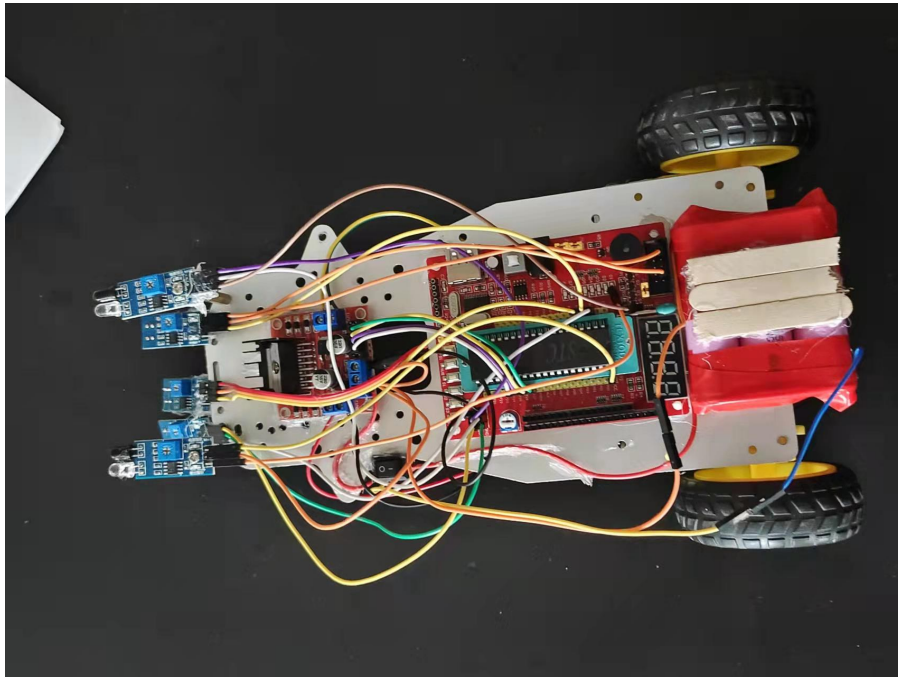
如若不修改电机转速的话，全功率下的转速是很快的，而对于本项目来说，小车速度并不能太快，因为要检测当前前方的黑线以障碍，如果速度太快，而运算速度跟不上或者考虑到数据传输本身有延迟。可能会出现响应信息还未到达就以及撞上去或者超出范围的情况，所以我们需要减速。而减速我们采取分片的方式来减速，比如当 speed 等于 7 的时候，0 ~ 7 有 8 个状态，速度也就会降低到原来的 1/8。

而最终经过调试，最终得出：

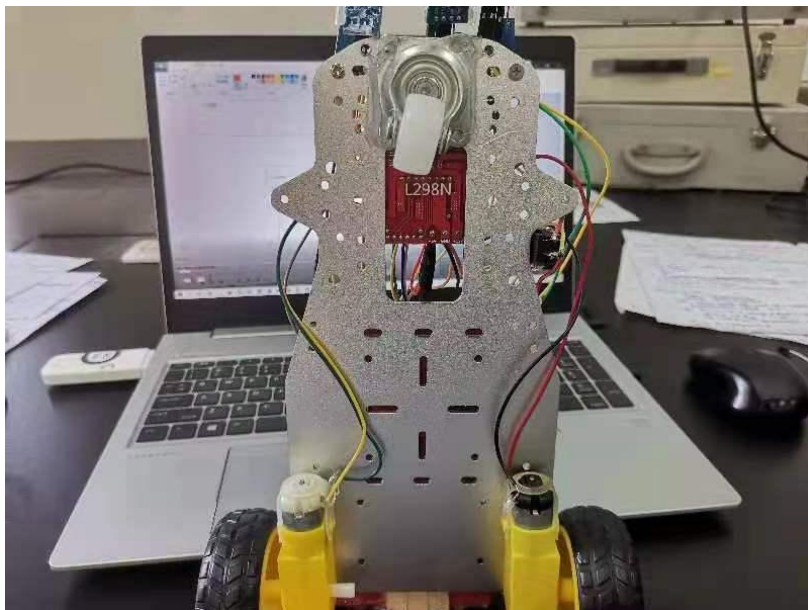
```
void go(char speed){  
for(char a = 0; a < speed; ++a){  
carStop  
}  
leftWheelFront  
rightWheelFront  
}
```

5.2 硬件调试

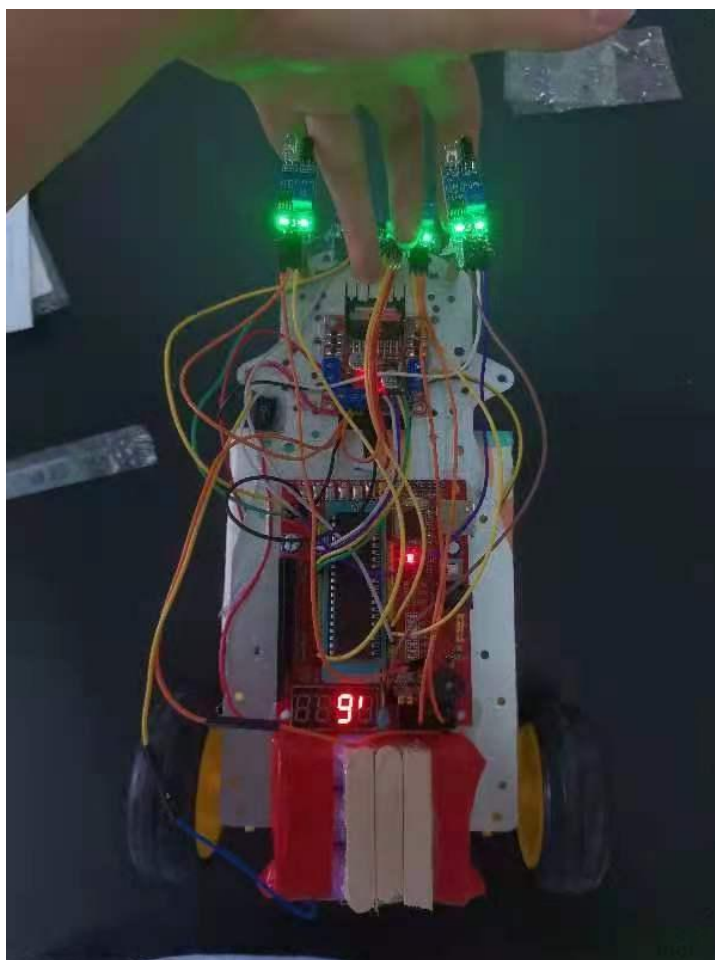
5.2.1 杜邦线的连接



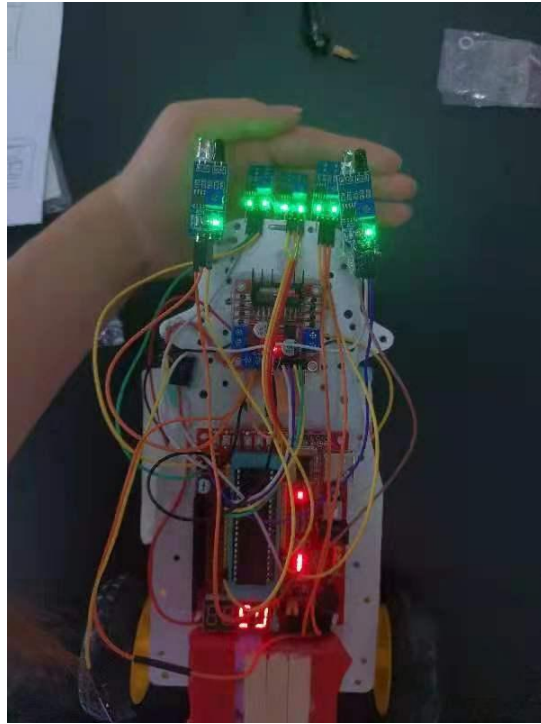
5.2.2 电机的固定以及测试



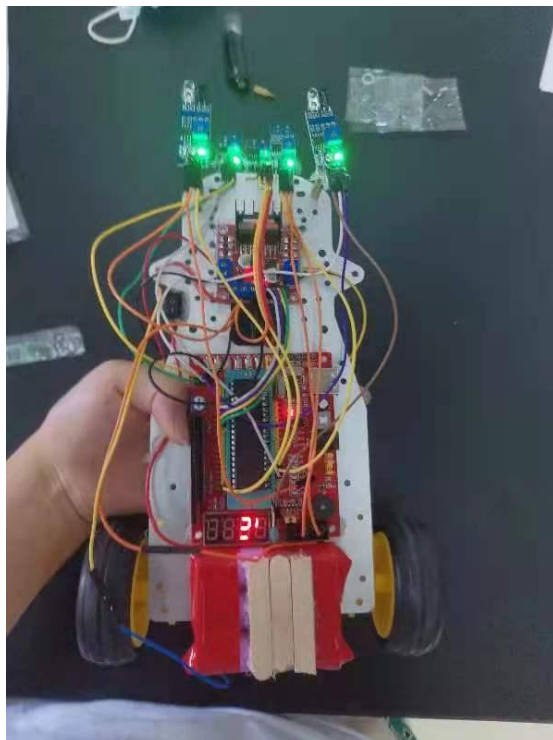
5.2.3 避障模块调试



5.2.4 TCRT5000 测试



5.2.5 系统总体运行图



5.3 系统总体性能分析

采取三个并排的 TCRT5000 作为检测口，极大的减少误差，正常情况下，处于中间的 TCRT5000 需要一直扫描到黑线，然后左右两侧的 TCRT5000 是一直扫描不到黑线的，但当中间的 TCRT5000 扫描不到，那么就势必左右两侧会有一个可以扫描到，若左侧扫描到黑线，说明当前需要右转，相反，若右侧扫描到黑线，说明当前需要左转。经过测试，该方案可以有效解决寻线问题。

而在小车车头放置的两个避障模块，分别处于左前方与右前方，当前方有障碍物的时候，避障模块可以迅速给单片机传播低电平，单片机收到低电平后开始停车，并开始报警和点亮报警灯。而选择左右两侧同时放置避障模块，可以有效避免仅放置一个避障模块的缺陷，可以完整辐射 180 度的正前方。也可以完美解决避障的问题。

但还是存在不足的地方，比如在理论上三个 TCRT5000 都检测不到黑线是不存在的情况，也就是说在小车正常运行的情况下，稍微移动是不会出现三个 TCRT5000 同时检测到黑线或同时检测不到黑线的情况的，但实际上，这种情况是存在的，造成这样的原因很多，比如代码方面不够严谨，有些情况没有考虑到，物理上三个 TCRT5000 的角度无法完整辐射正前方，再比如转弯的时候，是选择一轮后转一轮前转，还是一轮不动一轮前转的方式，以及地板的光滑度等等都影响因素。可以通过不断调整 TCRT5000 的角度，以及调节电位器来调整有效距离。优化代码以进行系统优化。

6 设计总结

本设计选用 C51 作为小车的主控版，借助 L298N 对单片机和部分模块供电，选择 L198N 之前，直接使用单片机给电机供电，发现单片机的正负给电机供电常，但引脚供电却使得电机无法正常运行，最终了解到单片机引脚虽然输出电压足够，但输出电流很小，远远达不到电机的驱动电流。这也就是说，要么电机一直转要么一直不转，做不到控制的目的。

而后想到二极管作为开关，于是将二极管接在电源与电机中间，另一端使用单片机，但也失败了。后了解到电机是有专门的驱动模块的，而后才选择使用 L298N 作为最终的电机驱动模块，而后遇到了第二个大问题，避障，如何让小车可以检测到前方的障碍物？有很多备选的方案，超声波，激光，红外，光敏等等。

超声波侧重测距，而非检测障碍，但也可以检测，不过超声波模块需要计算出小车前面的距离，来判断是否会撞到，对于本项目来说，没有显示前方距离的要求，暂时作为一个备选项。而光敏电阻侧重对光的敏感性，检测光亮程度，如果说障碍物以光线的形式给出，那么就需要光敏来作为避障元件。激光发射的话缺乏接受装置，故也不是最佳选项，最终选择了红外避障作为避障元件。

而检测黑线，则首选了 TCRT5000，这是毋庸置疑的，因为只有这个元件可以实现对黑线的检测。过程曲折，调试过程中很容易遇到电压不稳，杜邦线松动，跳线帽掉落的情况，不过算是打通了软硬件，学习了很多模块。同时也发现了自己诸多不足，一些基础知识掌握欠佳，还需不断完善和努力。

参考文献

- [1] <https://blog.csdn.net/wlswls1711/article/details/103113626>
- [2] <https://news.eeany.cn/eewiki/dzjczs/777597-21.html>
- [3] 李华. MCS-51 系列单片机使用接口技术 [M], 北京航空航天大学出版社, 1993.

附录

附录 1 源码

```
//SDCC 4.0.0 #11528 (Linux) + python3.9.7
#include <8052.h>
#define leftWheelFront P2_4=1;P2_5=0;
#define rightWheelFront P2_6=1;P2_7=0;
#define carStop P2_4=1;P2_5=1;P2_6=1;P2_7=1;
#define carTurnRight P2_4=0;P2_5=1;P2_6=1;P2_7=0;
#define carTurnLeft P2_4=1;P2_5=0;P2_6=0;P2_7=1;
#define leftUp P1_0
#define rightUp P1_1
#define leftDown P1_2
#define rightDown P1_3
#define center P1_4
#define sound P2_0
#define alertLed P1_0
int count = 0;
int time = 0;
char i = 0;
char table[]={0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82,
0xf8, 0x80, 0x90};
char End = 0;
// car go front
void go(char speed) {
for(char a = 0; a < speed; ++a) {
carStop
}
leftWheelFront
rightWheelFront
}
// alert sound
```

```

void Sound() {
    sound = 0;
    for(char a= 0; a <= 10; ++a);
    sound = 1;
}

// alert Led
void AlertLed() {
    alertLed = 0;
}

void interrupt_() __interrupt 1{
    TH0 = 0xFC;
    TL0 = 0x67;
    ++count;
    if(leftUp == 0 || rightUp == 0){
        Sound();
        AlertLed();
    }else
        alertLed = 1;
    if (count >= 1000){
        count = 0;
        if(!End)
            time++;
    }
    switch (i){
        case 0: P2_0 = 1; P0 = table[time % 10]; P2_1 = 0; ++i;
        break;
        case 1: P2_1 = 1; P0 = table[time % 100 / 10]; P2_0 = 0; --i;
        break;
    }
}

void main() {
    //open interrupt
    TMOD = 0x01;
    ET0 = 1;

```

```

EA = 1;
// set base data
TH0 = 0xFC;
TL0 = 0x67;
TR0 = 1;
while(End == 0) {
    if(center == 0 && rightDown == 0 && leftDown == 0 ) {
        End = 1;
        while(1) {
            carStop
        }
    }
    else if(center == 1 && rightDown == 1) {
        carTurnLeft
        for(char a = 0; a <= 10; ++a) {
            carStop
        }
    }
    else if( center == 1 && leftDown == 1) {
        carTurnRight
        for(char a = 0; a <= 10; ++a) {
            carStop
        }
    } else
        go(7);
}
}

```

附录 2 最小系统版原理图

