

电子技术课程设计总结报告

张博涵-智能 231-2023307150204

2025 年 8 月 5 日

目录

1 设计任务与器材清单	3
2 电路设计	3
2.1 电机控制电路设计	3
2.1.1 采用方案：使用 9110H 构造双 H 桥电路	3
2.1.2 备选方案：使用 L298N 构造双 H 桥电路	6
2.2 红外寻迹电路设计	6
2.2.1 版本一：双传感器寻迹	6
2.2.2 版本二：四传感器寻迹	6
2.3 电源稳压电路设计	6
2.4 车轮测速方案	7
2.4.1 备选方案：霍尔传感测速	7
2.4.2 采用方案：光电门测速	8
2.5 数码显示方案	9
2.5.1 采用方案：OLED 显示	9
2.5.2 备选方案：数码管 + 译码电路	9
2.6 声控电路	10
3 软件设计	10
3.1 寻迹算法	10
3.2 黑线检测与计圈逻辑	13
3.3 声控电路软件实现	15
3.4 完整代码逻辑	17
4 电路组装与调试	18
4.1 调试红外寻迹	18
4.2 调试声控电路	18
4.3 点位检测	19
4.4 整车拼装焊接	20

5 总结	20
6 参考文献	20

1 设计任务与器材清单

项目旨在设计并实现一套智能循迹小车系统，通过软硬件协同完成的控制与检测任务。具体的设计任务和功能要求如下：

基本要求

1. **小车循迹**：小车能够识别并沿着预先设定的轨道（黑色轨迹）自动行驶，实现自动循迹控制。
2. **规定动作**：小车需从 A 点出发，沿轨道连续行驶 2 圈后准确停车，用时越短越好，以体现系统响应速度和稳定性。
3. **计时显示**：系统应具备计时功能，能够实时显示小车的行驶时间（计时范围 0-99s），停车后自动显示总行程所用时间，便于评估任务完成效率。
4. **避障功能**：当小车行驶过程中遇到前方障碍物时，能实现紧急刹车并启动声光报警提示，障碍物清除后可自动恢复正常行驶，提升系统的安全性和智能化水平。

提高部分

1. **车速控制**：小车行驶速度可实现无极调节，便于适应不同场景和需求。
2. **附加动作**：小车从 A 点出发，沿轨道行驶 2 圈，在特定弯道（如 BC 段、DA 段）自动减速通过，并在经过 A 点时停车 10 秒，每经过一个关键点均有声光提示。
3. **里程检测**：系统能够实时检测并显示小车的累计行驶里程，便于全过程监控和数据分析。
4. **车速检测**：小车行驶过程中可实时检测并显示当前车速，为调试和优化提供数据支持。
5. **声控功能**：通过声控模块（如拾音麦克风等），可用声音信号控制小车的起停动作，进一步提升系统的智能交互体验。

为保证实验系统的功能完整性与可靠性，项目所有器件均根据实验设计需求及实际性能进行筛选，能够满足系统各功能模块的搭建、测试与后续扩展。数量统计如表 1 所示。

2 电路设计

2.1 电机控制电路设计

2.1.1 采用方案：使用 9110H 构造双 H 桥电路

众所周知，电机两端电压的方向与大小决定了电机的运动状态，因此常采用桥式开关电路来控制电机的转动方向。通过 CMOS 门控器件进行逻辑编程，可以以最少的开关数量同时控制电机的正反转及启停。该电路结构被称为 H 桥，只需调整输入电压的占空比（PWM）即可进一步调节输出电压幅值，从而实现车速的控制。

L9110H 是一款集成单个 H 桥的驱动芯片，而 L298N 则在一颗芯片内部集成了两个 H 桥电路。因此，在双轮驱动的小车上，可以采用两块 L9110H 芯片分别控制左右轮电机的启动、停止及转向。IB、IA 引脚直接连接 STM32 的 PWM 输出，有助于简化整体线路布局，提高控制的灵活性和可维护性。相关电路的简化原理图与仿真图如图 1 和图 2 所示。

表 1: 实验器材清单

器材名称	规格/备注	数量
小车底盘	含车轮、减速直流电机、测速码盘等	2
STM32 开发板	含 ST-LINK 仿真器、Type-C 下载线	1
200Ω 电阻	±5%, 1/4W	2
1K 电阻	±5%, 1/4W	5
10K 电阻	±5%, 1/4W	10
100K 电阻	±5%, 1/4W	1
10uF 电容	陶瓷, 直插	2
10K 多圈精密电位器	3296W	5
LM393	双列直插	7
LM7805	直插 TO-220	1
L9110H	双列直插	2
2N9013	直插 TO-92	2
槽型光耦 ITR9606		1
红外传感 TCRT5000		5
拾音麦克风 (咪头)	电容式, 56dB, 6*5mm	1
PCB 实验板 (洞洞板)	5 连孔, 单面 100×150mm	6
18650 电池	锂电池 3.7/4.2V	2
发光二极管	红、绿、黄、白	3
蜂鸣器	电磁式, 无源	1
DIP 芯片插座	对应数量的不同 DIP 芯片插座	若干

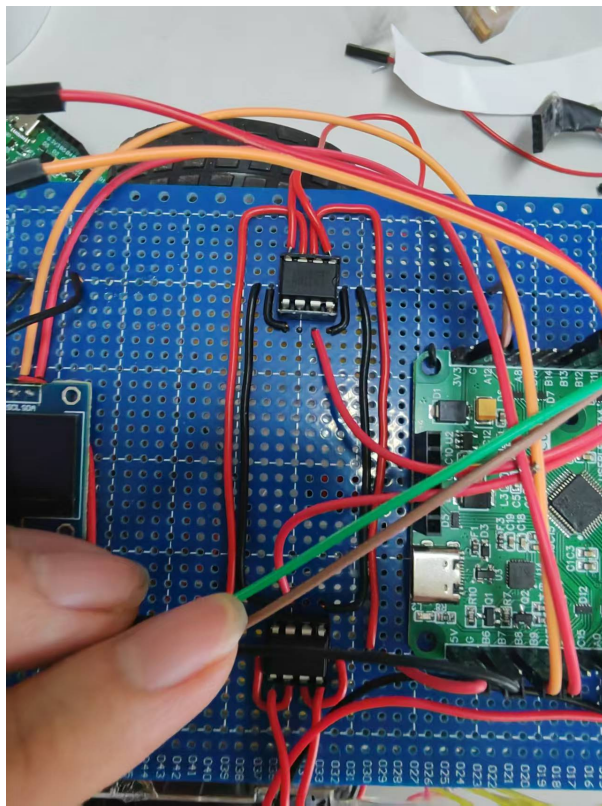


图 1: H 桥电路实物图

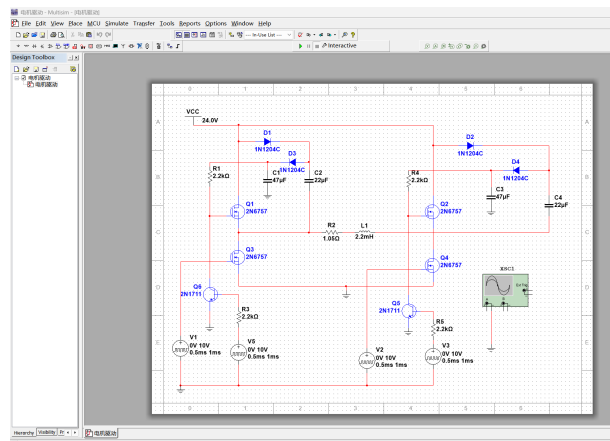


图 2: H 桥电路仿真图

2.1.2 备选方案：使用 L298N 构造双 H 桥电路

作为备选方案，也可以采用 L298N 芯片实现双 H 桥驱动电路，其原理与 L9110H 方案基本一致。但综合考虑到 L9110H 芯片布局更为紧凑、走线更加简便，最终本设计未采用 L298N 方案，仅作为可行的技术储备。

2.2 红外寻迹电路设计

2.2.1 版本一：双传感器寻迹

本节首先介绍红外寻迹模块的设计，其仿真电路如图所示。红外模块的主要作用是检测光信号的变化，本电路利用比较器的高增益特性，将微小的光敏变化“放大”为数字开关信号，输出结果由 VR1 与 10K 电阻组成的分压网络设定门槛。红外接收端的分压由光敏管的电阻决定：受光时电阻变低，导致 V_+ 降低；无光时电阻变高， V_+ 升高。正输入端的电压由 VCC 通过 10K 和 VR1 (10K) 串联分压获得：

$$V_- = V_{CC} \times \frac{VR1}{10K + VR1} \quad (1)$$

其中，VR1 的调节范围为 $0 \sim 10K\Omega$ ，因此 V_- 可在 $0 \sim V_{CC}$ 范围内调节，通过旋转 VR1 可以实现阈值的连续调节。负输入端的电路结构为：上拉 10K 串联光敏管（红外对管接收端），下拉至地，同时通过 104 电容进行滤波，其分压表达式为：

$$V_+ = V_{CC} \times \frac{R_{IR}}{10K + R_{IR}} \quad (2)$$

其中 R_{IR} 为光敏三极管的等效电阻，受光强影响极大，通常变化范围为几百欧姆至几兆欧姆。若有红外反射（受光）， R_{IR} 很小， V_+ 接近地电位；若无红外反射（无光）， R_{IR} 很大， V_+ 接近 V_{CC} 。电路设计与实物焊接图如图 3 和图 4 所示。

2.2.2 版本二：四传感器寻迹

在实际应用过程中发现，双传感器结构在某些极端情况下无法保证小车稳定寻迹，尤其是在急转弯时容易偏离轨道。为提升系统的鲁棒性和轨迹捕捉能力，我们在两侧增设了额外的两个红外传感器，构成四传感器寻迹系统，其电路结构与前述完全一致。

2.3 电源稳压电路设计

本设计采用 LM7805 线性稳压芯片作为电源稳压方案，如图所示。稳压模块是电路中不可或缺的部分，主要基于以下两点考虑：首先，芯片及其他分立器件的正常工作电压为 5V，而实际电源通常由电池提供，输出电压可达 7V 及以上，因此必须通过稳压模块将高电压转换为稳定的 5V 供电。其次，在实际调试过程中发现，如果直接使用电压源为系统供电，随着电机等大功率负载的接入，系统电压容易出现波动，尤其是在高速运行时，电机产生的干扰会导致计数器等关键模块工作不稳定，而在低速运行时影响较小。这一现象说明电源稳定性对整个系统的可靠性至关重要，因此稳压电路的加入是非常必要的。本设计中，选用经典的 LM7805 稳压芯片，并在其输入端和输出端分别接入了不同类型和容量的电容，以提升电源品质。其具体作用如下：

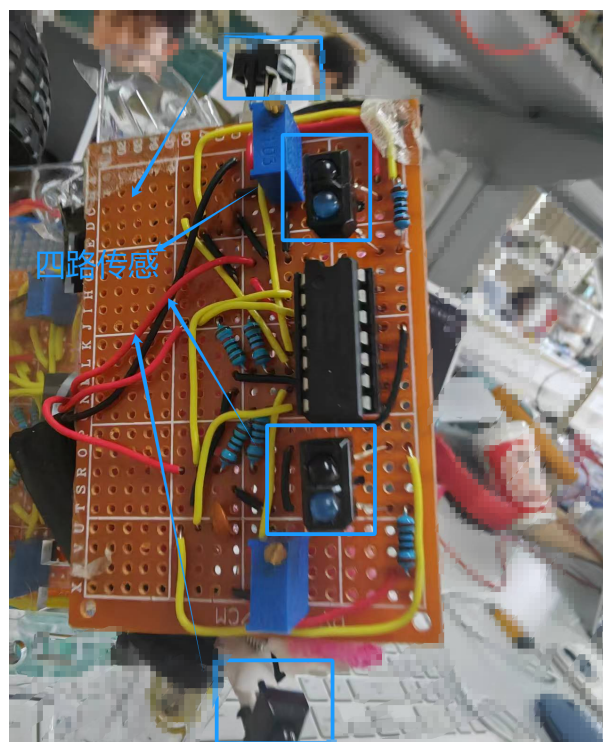


图 3: 红外寻迹电路实物图

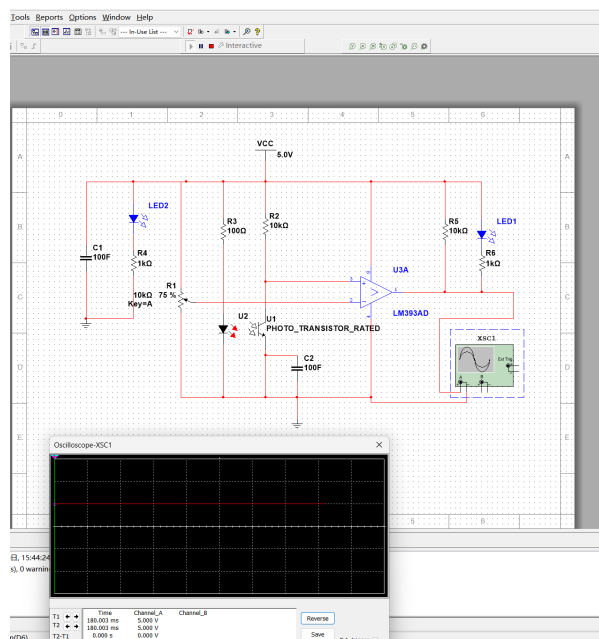


图 4: 红外寻迹电路仿真图

- **C2 (1000 μ F, 电解电容)**: 位于 LM7805 输入端, 用于对输入电源进行滤波和平滑, 吸收因电源线路阻抗或负载切换引起的电压尖峰和纹波, 为稳压芯片提供较为纯净的直流输入电压。
- **C1 (1 μ F, 电解电容)**: 并联于 LM7805 输出端, 用于进一步抑制输出电压的高频噪声和微小纹波, 提高稳压器响应速度和稳定性。
- **C4 (220 μ F, 电解电容)**: 同样并联在输出端, 容量较大, 主要用于对负载突变时的电流需求进行补偿, 降低因大电流冲击引发的输出电压波动, 从而确保后续电路的正常供电。

整体稳压电路结构如图 5 所示,

2.4 车轮测速方案

2.4.1 备选方案: 霍尔传感测速

备选方案使用霍尔测速方案, 事实上我们首先采用并搭建的方案也是这一方案, 我们在车轮上粘上磁铁块, 直接使用霍尔元件感应磁场周期性变化, 然后从引脚接受信号返回单片机。经过查阅资料与调试, 得知霍尔元件输出的是一个微小模拟变化量, 因此我们利用设计好的放大电路, 对信号进行放大检测, 电路具体结构与仿真演示在下图展示了出来。

但是调试过程中遇到了两个问题, 经过分析弃了此方案, 选取了光电测速的思路。问题一是, 我们发现霍尔元件对磁场较为敏感。除了车轮上固定的磁铁以外, 其他磁场都会对霍尔元件的检测起到影响, 因此产生波形噪声很多, 需要很强大的物理滤波电路或者代码滤波算法。问题二: 霍尔元件及其模

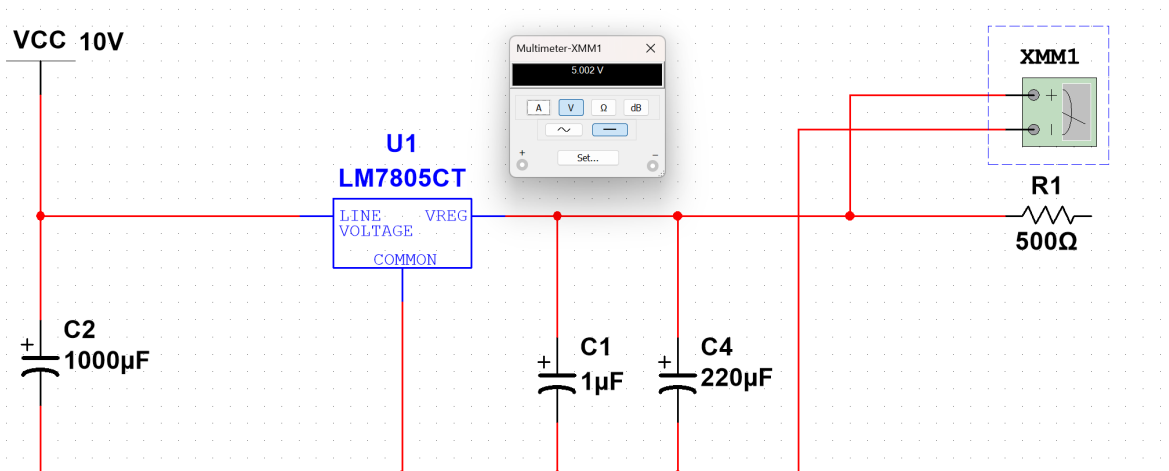


图 5: LM7805 稳压电源电路

块很难以合适的角度固定在车底板附近，给实际工程带来了很大的困扰。综上所述，我们放弃了霍尔元件的方案。

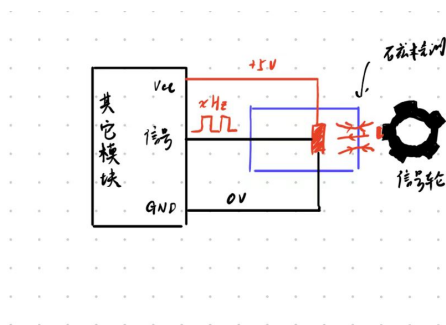


图 6: 霍尔测速示意图

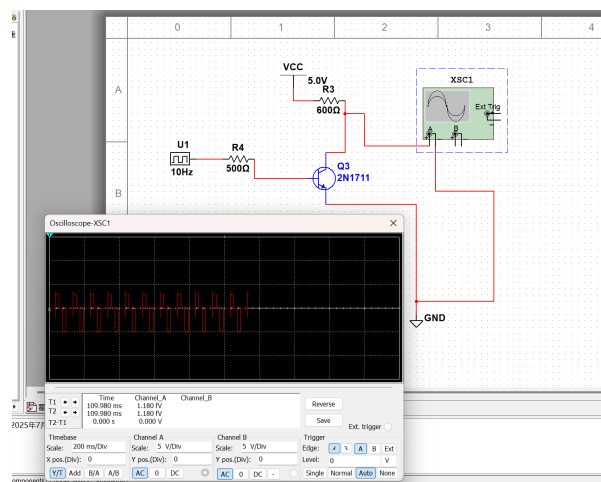


图 7: 霍尔测速仿真图

2.4.2 采用方案：光电门测速

经过方案比选，最终我们采用了光电门测速的方式实现小车轮速的精准测量。光电门测速方案结构简单，响应速度快，适合在小车底盘上应用。其原理如下：在电路中，发光二极管（LED）作为光源，持续向对面的光敏元件（如光敏三极管或光敏电阻）发射光线。当车轮上的遮光片（或测速码盘）转动至光路之间时，会间断性地遮挡光线，造成光敏元件接收到的光强周期性变化。此时，光敏元件的电阻随光照强度发生变化，进而影响后级分压点的电压。当遮挡发生时，分压点的电位升高或降低，经过电容滤波和电阻分压后，将此微小的电信号变化送入比较器（LM393）的输入端。比较器将输入信号与设定阈值电压进行比较，当检测到信号超过阈值，输出清晰的高、低电平脉冲信号。这一输出既可直接驱

动 LED 指示灯进行可视化提示，也可被单片机采集，用于实现车轮的测速与计数功能。通过测量脉冲的时间间隔或单位时间内脉冲数量，即可准确计算出车轮的转速和小车的运行速度。

图 8展示了本项目实际使用的光电门测速电路的原理图：

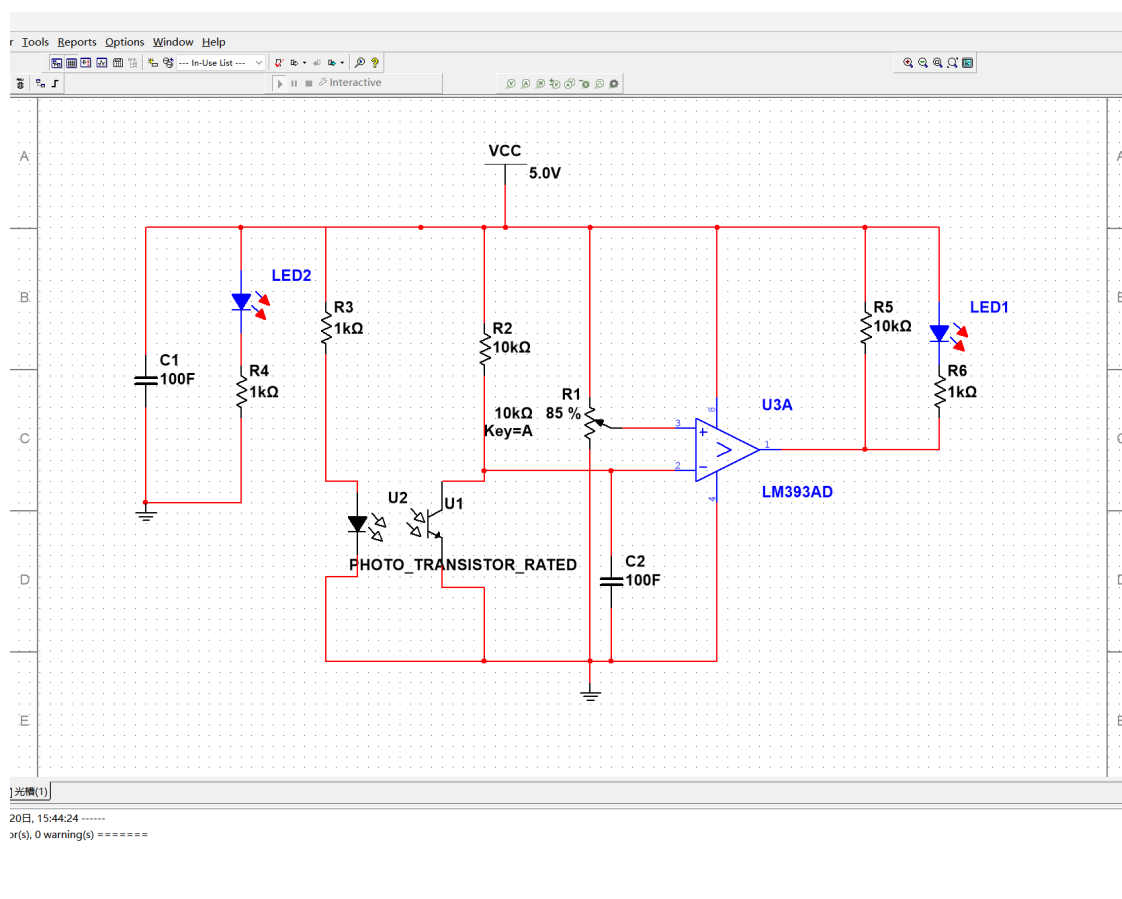


图 8: 光电门测速原理图

2.5 数码显示方案

2.5.1 采用方案：OLED 显示

在本项目中，数码显示模块承担着计时、速度、里程的显示任务。经过综合评估，最终选择了 OLED 显示屏作为数码显示方案。OLED 显示屏具体积小、功耗低，并且支持灵活的字符显示。相比传统的数码管方案，OLED 可通过 I²C 或 SPI 等总线与主控芯片连接，占用极少的 I/O 引脚资源因此，本系统直接通过编程将相关数据输出到 OLED 显示屏。

2.5.2 备选方案：数码管 + 译码电路

我们准备了数码管 + 译码电路作为备选，但考虑到如下三点缺点最终放弃此方案。缺点一：数码管一个元件只能显示一位数字，我们的任务需要计时显示，里程显示，速度显示三种显示类型，即使使用高速切换 IO 扫描的方案也比较繁琐。缺点二：针对性设计大量译码电路难度较高，同时接线与焊接

复杂，耗时间，容易出错。其三：过多的元件占用空间。对于本就缺少空间的小车而言更甚。我们的备选方案以及电路设计如下，我首先用基本的数电模型设计了右边的电路，然后查阅器件清单表，发现有很多集成模块可以直接使用，最终，为了尽可能节省引脚，我设计了左图手绘电路作为备用方案。

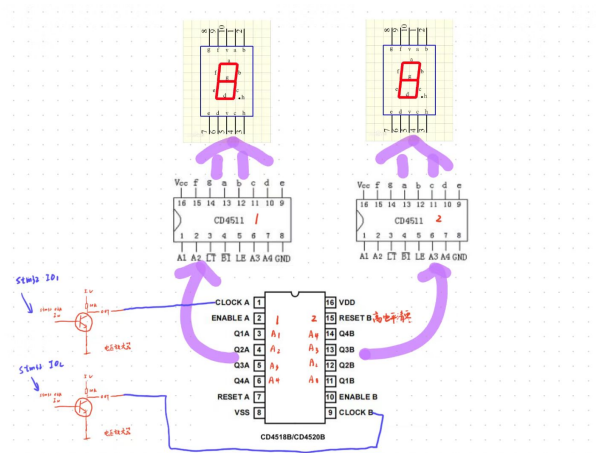


图 9: 数码管电路示意图

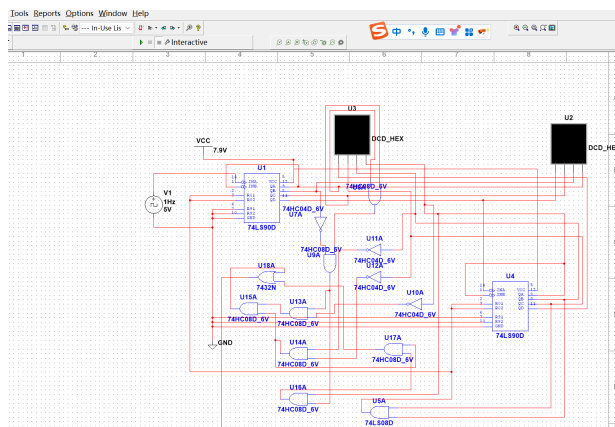


图 10: 数码管电路仿真图

2.6 声控电路

声控电路检测环境中的声音变化，当外界产生声响时，麦克风（咪头）会捕捉到微小的电信号，该信号首先经过电容耦合，输入到二级放大电路。经过两级放大后，输出的模拟信号幅度大大提升，可被 STM32 等主控芯片采集用于后续判别和控制。在初始设计中，第一级放大电路的基极偏置电阻采用 $1\text{ M}\Omega$ ，此时电压增益大于 2000 倍，麦克风对外界声音极为敏感。由于第一级无发射极退化电阻，静态电流较小，晶体管工作点接近放大区边缘，极易被微小的声压信号（如基极 2 mV 级别的扰动）触发，导致输出节点出现数伏的跳变，电路容易对环境噪声及微弱扰动产生误判，不利于后级的稳定阈值判断。

为优化电路性能，经过多次实验调整，最终将基极偏置电阻改为 $100\text{ k}\Omega$ 。调整后，第一级静态工作点推入饱和区，只有当输入声压足够大时，基极电压才会下降使三极管暂时退出饱和状态，进而触发第二级导通，输出从近 0 V 跃升至约 1.7 V （对应 LED 导通电平）。这样，电路从“线性高增益”模式转变为“阈值/开关”模式，只有达到一定声强才输出明显的高电平，有效降低了对环境噪声的误触发概率，非常适合声控逻辑判断与开关控制应用。本次设计的声控电路手绘原理图及其仿真效果如图 11 和图 12 所示。

3 软件设计

3.1 寻迹算法

小车循迹的核心在于对传感器信号的实时采集与决策。不同数量的传感器配置，对循迹精度和容错能力有显著影响。为实现循迹功能，本系统分别设计了“两传感器”和“四传感器”两种控制算法，具体流程如下：

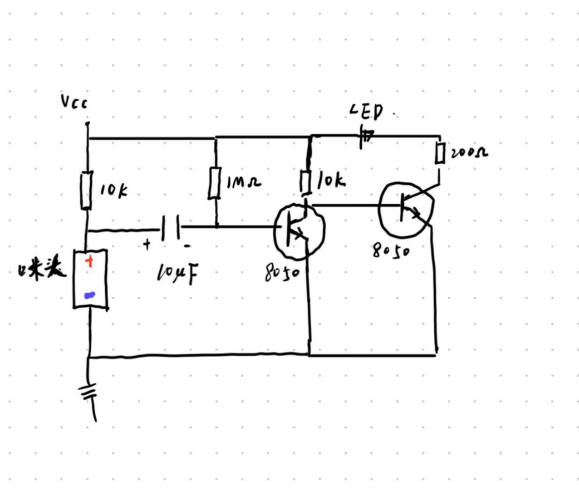


图 11: 声控电路手绘原理图

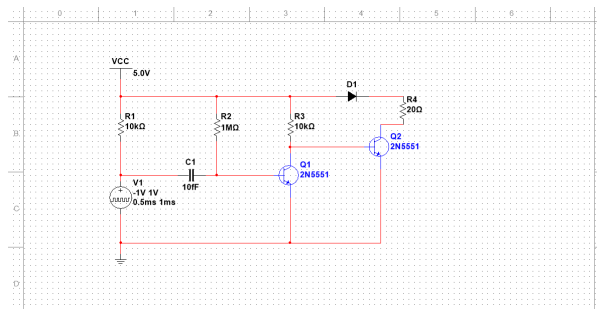


图 12: 声控电路仿真图

(1) 两传感器循迹算法

该方案在小车前端左右各布置一个红外传感器，通过判断两侧传感器是否检测到黑线，决定小车前进、左转或右转。若两传感器均未检测到轨迹，则小车直行；若左、右某一侧传感器偏离轨迹，则小车自动修正方向。流程如算法 1所示。

(2) 四传感器循迹算法

为提升循迹鲁棒性和转弯精度，在小车前端两侧各加装一组红外传感器，左右各两个。通过对左右两侧的传感器信号进行“或”运算，扩大检测区域，提高判断灵敏度和可靠性。流程如算法 2所示。

Algorithm 1: 两传感器控制流程**Input:** 左传感器, 右传感器**Output:** 小车运动方向与状态显示

```

while True do
    left ← 读取左传感器信号;
    right ← 读取右传感器信号;
    if left == 0 and right == 0 then
        前进 (Car_Forward) ;
        timing_enabled ← 1;
        OLED 显示"Running";
    else
        if left == 0 and right == 1 then
            左转 (Car_Left) ;
            timing_enabled ← 1;
            OLED 显示"L-slow-down";
        else
            if left == 1 and right == 0
            then
                右转 (Car_Right) ;
                timing_enabled ← 1;
                OLED 显
                示"R-slow-down";
            else
                前进 (Car_Forward) ;
                timing_enabled ← 1;
                OLED 显示"On Mark";
            end
        end
    end
end
end

```

Algorithm 2: 四传感器控制流程**Input:** 左一, 左二, 右一, 右二传感器**Output:** 小车运动方向与状态显示

```

while True do
    left1, left2 ← 读取左一, 左二传感器;
    right1, right2 ← 读取右一, 右二传感器;
    left ← left1 or left2;
    right ← right1 or right2;
    if left == 0 and right == 0 then
        前进 (Car_Forward) ;
        timing_enabled ← 1;
        OLED 显示"Running";
    else
        if left == 0 and right == 1 then
            左转 (Car_Left) ;
            timing_enabled ← 1;
            OLED 显示"L-slow-down";
        else
            if left == 1 and right == 0
            then
                右转 (Car_Right) ;
                timing_enabled ← 1;
                OLED 显
                示"R-slow-down";
            else
                前进 (Car_Forward) ;
                timing_enabled ← 1;
                OLED 显示"On Mark";
            end
        end
    end
end
end

```

上述两种算法均采用循环判断传感器状态, 并实时调整小车运动方向, 同时通过 OLED 屏幕反馈状态信息。两传感器方案结构简单、便于实现, 但在复杂轨迹或快速转弯时容错性略低; 四传感器方案有效提升了循迹鲁棒性。

3.2 黑线检测与计圈逻辑

为了准确统计小车越过终点线（黑线）的次数，以及实现多点位的声光提示与定圈停车功能，本设计采用了“状态沿”检测和“循环计数”结合的方式。

1. 如何避免重复计数

在实际检测过程中，如果仅依靠传感器电平判断，会出现“小车停留或颠簸于黑线上时连续多次计数”的问题。为避免这一误计现象，本方案维护了两个关键变量：**at_mark**（当前是否在黑线上）和**last_at_mark**（上一个时刻是否在黑线上）。只有在**at_mark**由0变为1的“上升沿”瞬间，才将计数变量**count**加一。这样，无论轮子如何抖动或小车如何慢速通过黑线，都只会被计一次，实现了高可靠性的黑线穿越检测。

2. 圈数记录与多点提示

为实现“完成两圈行驶”“经过 A/B/C/D 四个关键点声光提示”等功能，算法将每圈分为四段，每段以一条特定黑线作为分界。采用计数变量**count**记录每次穿越黑线的累计次数。由于一圈有四个关键点，当

$$count \bmod 4 = 0$$

表示经过 A 点； $count \bmod 4 = 1$ 表示经过 B 点，依次类推。这样，无需复杂状态机即可实现对每一圈、每一个关键点的准确声光提示及停车逻辑。基于上述原理，黑线检测与计圈停车的完整流程如算法 3 所示：

Algorithm 3: 终点线检测与计圈停车逻辑**Input:** 左红外传感器 left, 右红外传感器 right**Output:** 完成两圈后停车并显示初始化变量: $\text{count} \leftarrow 0$, $\text{stopped} \leftarrow 0$, $\text{last_at_mark} \leftarrow 0$;

```

while True do
    if stopped == 1 then
        停车;
        计时停止;
        显示"Finish 2 laps";
        蜂鸣器关闭;
        跳出循环, 程序停止
    end
    else
        蜂鸣器关闭;
        读取 left, right 传感器;
         $\text{at\_mark} \leftarrow (\text{left} == 1 \text{ and } \text{right} == 1) ? 1 : 0$ ;
        if  $\text{last\_at\_mark} == 0$  and  $\text{at\_mark} == 1$  then
             $\text{count} \leftarrow \text{count} + 1$ ;
        end
         $\text{last\_at\_mark} \leftarrow \text{at\_mark}$ ;
        if  $\text{count} \geq 8$  then
             $\text{stopped} \leftarrow 1$ ;
            停车;
            显示"Finish 2 laps";
            跳过后续流程, 程序结束
        end
        if  $\text{left} == 0$  and  $\text{right} == 0$  then
            前进; 显示"Running";
        else
            if  $\text{left} == 0$  and  $\text{right} == 1$  then
                左转; 显示"L-slow-down";
            else
                if  $\text{left} == 1$  and  $\text{right} == 0$  then
                    右转; 显示"R-slow-down";
                else
                    前进; 显示"On Mark";
                end
            end
        end
    end
end

```

3.3 声控电路软件实现

声控功能依托于麦克风前端放大电路和 STM32 内置 ADC 模块。通过采集麦克风输出的电压信号，并结合阈值判断与边沿检测，实现了用“拍手”或短促声音控制小车起停的逻辑。

1. ADC 信号采集与电压转换

首先，系统初始化 STM32 的 ADC 模块，将与声控放大电路连接的引脚（如 PA1）设置为模拟输入。通过定时采样，获取麦克风电路输出的瞬时电压值，并归一化为标准电压。电压的计算方式如下：

$$V = \frac{\text{ADC 读取值}}{4095} \times V_{\text{REF}}$$

其中 $V_{\text{REF}} = 3.3\text{V}$ 为参考电压。

2. 边沿检测与声控判定

为防止因持续噪声或多次抖动导致误触发，程序引入 `sound_flag` 状态变量，配合阈值法实现“上升沿”检测。具体流程为：当 ADC 采集电压低于 0.6V ，且此前 `sound_flag` 为 0 时，判定为有效触发，并将 `sound_flag` 置为 1，防止多次响应。只有当电压回升至 0.6V 及以上时，才允许再次触发。

3. 起停控制逻辑

小车的当前运动状态由 `car_state` 变量标记。每次声控被有效触发时，根据 `car_state` 的当前值，切换小车的前进与停止状态，并实时在 OLED 屏幕上显示对应的状态信息。

4. 主程序流程

主循环中，系统定期调用声控检测函数，并通过适当延时保证人耳可感知的反馈响应速度。主要实现流程如下：

- 初始化相关硬件模块（电机、OLED、ADC 等）。
- 在主循环中周期性调用 `sound_check()` 函数。
- 若检测到声控触发，切换小车状态，刷新 OLED 提示。
- 通过 `Delay()` 适当延时，避免刷新过快。

5. 代码逻辑核心片段说明

```
1 void sound_check()
2 {
3     float a = ADC_Read_Voltage(); // 读取麦克风输出电压
4     sprintf(disb_buf, "A=%.2fV", a);
5     OLED_ShowString(2, 1, disb_buf);
6
7     // 声控边沿检测，避免重复触发
8     if(a < 0.6 && sound_flag == 0)
9     {
10         sound_flag = 1;
11         if(car_state == 0)
12         {
13             Car_Forward();
14             OLED_ShowString(3, 1, "Run");
15             car_state = 1;
16         }
17     }
18     else
```

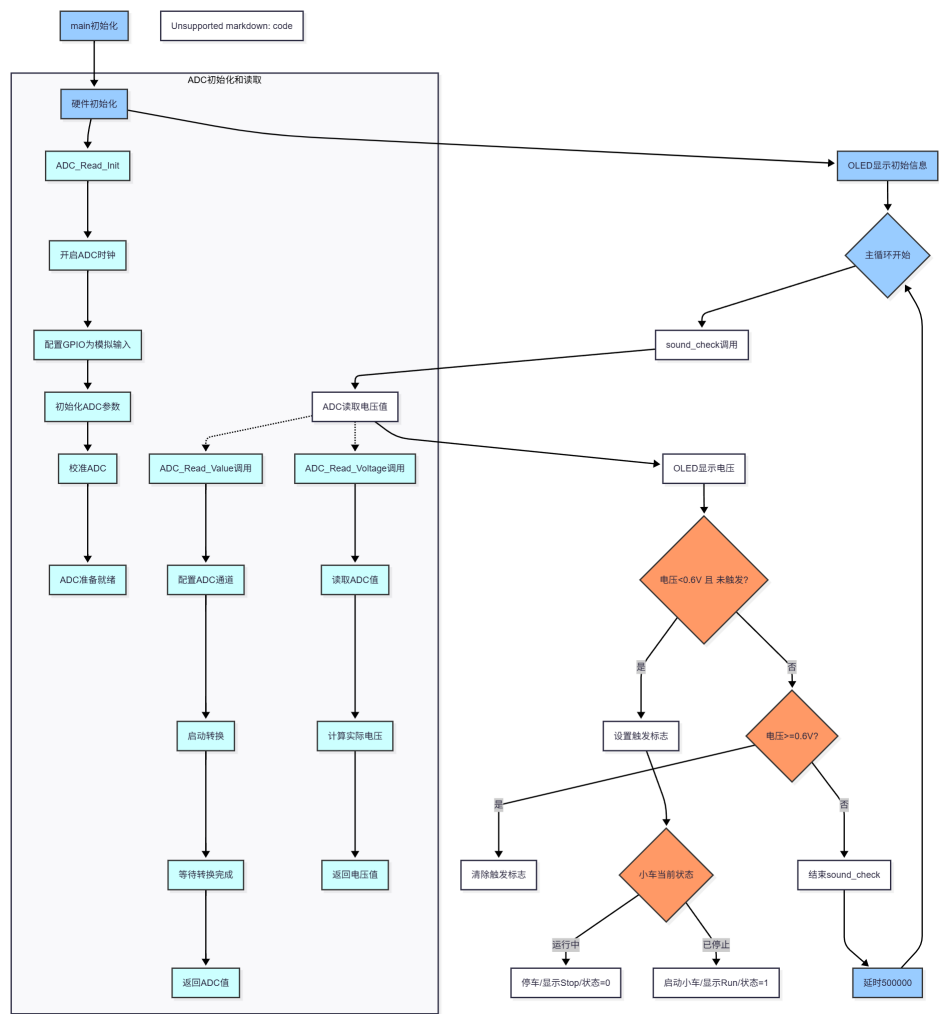


图 13: 声控代码流程

```
18 {
19     Car_Stop();
20     OLED_ShowString(3, 1, "Stop");
21     car_state = 0;
22 }
23 }
24 if(a >= 0.6)
25 {
26     sound_flag = 0; // 松手，允许下次触发
27 }
28 }
```

Listing 1: 声控触发及小车起停核心逻辑

如上所示，整个声控系统依赖“ADC 电压采集—阈值判定—状态切换”这一主线逻辑。能在实际环境下稳定实现“一声启动、一声停车”等智能控制需求。整体思路见图 13

3.4 完整代码逻辑

完整程序采用模块化设计思想，主控代码分为多个功能模块进行管理，主要是电机驱动、循迹识别、OLED 显示、计时模块、蜂鸣器控制、ADC 读取、声控等。主函数中包括初始化、主循环处理、判断逻辑、控制输出。

1. 模块初始化

程序开头依次调用如下初始化函数:Motor_Init() 初始化电机,OLED_Init() 初始化显示屏,Irtracking_Init() 初始化红外循迹模块,Speed_TIM2_Init() 和 Speed_EXTIO_Init() 设置测速功能,bizhang_Init() 启用红外避障模块。此外,使用 SysTick_Config() 实现 1ms 计时功能,为后续定时逻辑提供支持。

2. 主循环逻辑

主循环中,程序不断执行以下几项功能:

- **障碍检测优先级最高**: 使用 bizhang_get() 函数实时检测前方是否存在障碍物,若检测到障碍,小车立即停止,蜂鸣器报警并中断后续操作。
- **循迹运行控制**: 通过 Left_Irtracking_Get() 和 Right_Irtracking_Get() 获取左右红外传感器信号,判断当前小车是否在线上,并根据判断结果执行前进、左转、右转等指令。
- **计点 / 计圈逻辑**: 通过两个红外传感器同时检测到黑线判断是否经过标记点,每经过一个点即计数加一,经过固定点数后停止运行。第一段代码采用“过 8 个点停止并在第 4 个点暂停 3 秒”的策略,第二段代码则采用“完成 2 圈停止”的策略。
- **计时功能**: 当小车进入正常运行状态时,启用计时器,每秒钟在 OLED 上更新一次运行时间,范围为 0-99 秒。
- **实时信息显示**: OLED 屏显示当前运行状态(如“Running”、“On Mark”、“Obstacle”等)、运行时间、速度、累计行驶距离等,便于调试和观察。

3. 触发事件处理

在第一个版本中,小车每通过一个标记点会执行 BeepAndLed() 函数使蜂鸣器响起,并在第 4 个点额外暂停 3 秒;在第二个版本中,则更注重避障处理,小车检测到前方有物体则立即停下,直到障碍移除为止。

- 系统初始化
- 检测障碍物: 若有 → 停车报警
- 否则进入循迹模式
- 判断是否到达计数点
- 达到目标点数或圈数 → 停车
- 否则继续寻迹,并记录时间

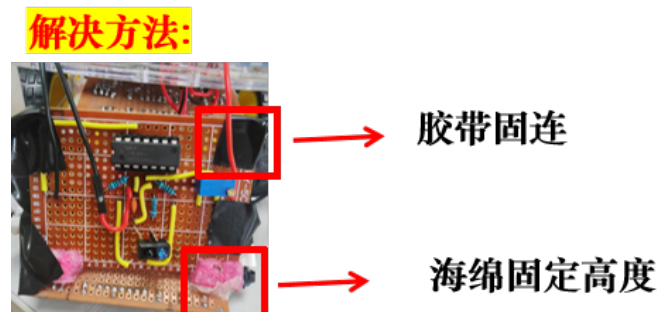


图 14: 高度固定方式

4 电路组装与调试

4.1 调试红外寻迹

在系统调试过程中，红外寻迹模块曾多次出现故障或异常

问题一：2025 年 7 月 15 日晚，模块半边失效原本工作正常的红外寻迹模块在晚间调试时，突然出现一侧传感器无响应的情况，导致小车运行异常。故障排查如下：

- 使用万用表依次检测模块各节点电压，发现对比正常侧，故障侧的 V- 电压明显异常，达到 4.1V。
- 初步排除光电管损坏、LM339 比较器失效、电阻、电容或电位器焊接不良等常见问题。
- 进一步分析发现，问题根源在于电位器的阈值设定偏高，导致 LM339 输出长时间为高电平，使得该通道误判为“非黑线”，从而始终不触发转向。

解决方法：重新调整电位器，使输出电压与实际地面反射光强对应阈值匹配，最终恢复正常工作。

问题二：2025 年 7 月 17 日深夜，小车识别黑线能力变弱在进行晚间例行测试过程中，发现原本灵敏的循迹系统识别能力明显下降，小车在黑线上频繁漂移，误判率大幅上升。排查流程如下：

- 首先复查电位器设定值，理论计算与实测均在合理范围内。
- 通过电压表测试传感器各关键节点电平，确认供电、GND、比较器输出等皆正常。
- 通过 OLED 显示屏输出传感器读取值，发现 MCU 接收数据也符合预期，基本排除软件逻辑问题。

最终定位原因：由于小车长时间行驶，可能多次经过地面凹陷或凸起，使得红外传感器与地面之间的相对距离和垂直角度发生偏移。即使电路和程序正常，若光电管倾斜或浮动，也会造成误判。**解决方法：**使用胶带将模块牢固固定，并在下方加入海绵垫以调整与地面之间的高度，确保传感器垂直朝向地面、并保持 8-12 mm 左右的相对距离。修正后系统稳定性显著提升。

4.2 调试声控电路

这一部分调整了第一级的上拉电阻，最终选取了合适的阻值 100 欧姆，模块可以正常使用。

4.3 点位检测

这一部分需要反复确定角度,并排查引脚电压是否正常。由于程序不能直接 print, 因此使用 OLED 屏幕进行调试,我当时调试的留存代码如下:

```
1 #include "stm32f10x.h"
2 #include "OLED.h"
3 #include "Irtracking.h"
4
5 int main(void)
6 {
7     OLED_Init();
8     OLED_Clear();
9     Irtracking_Init();
10
11     while (1)
12     {
13
14         uint8_t s1 = GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_10); // S1
15         uint8_t s2 = GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_11); // S2
16         uint8_t s3 = GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_12); // S3
17         uint8_t s4 = GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_13); // S4
18
19
20         // S1:0 S2:1 S3:0 S4:1
21         char disp_buf[20];
22         sprintf(disp_buf, "1:%d 2:%d 3:%d 4:%d", s1, s2, s3, s4);
23         OLED_ShowString(1, 1, disp_buf);
24
25
26         // OLED_ShowString(2, 1, "L1:");
27         // OLED_ShowNum(2, 4, s1, 1);
28         // OLED_ShowString(2, 6, "L2:");
29         // OLED_ShowNum(2, 9, s2, 1);
30         // OLED_ShowString(2, 11, "R1:");
31         // OLED_ShowNum(2, 14, s3, 1);
32         // OLED_ShowString(2, 16, "R2:");
33         // OLED_ShowNum(2, 19, s4, 1);
34
35         for (volatile int i=0; i<100000; i++);
36     }
37 }
```

Listing 2: 点位检测电平调试代码

4.4 整车拼装焊接

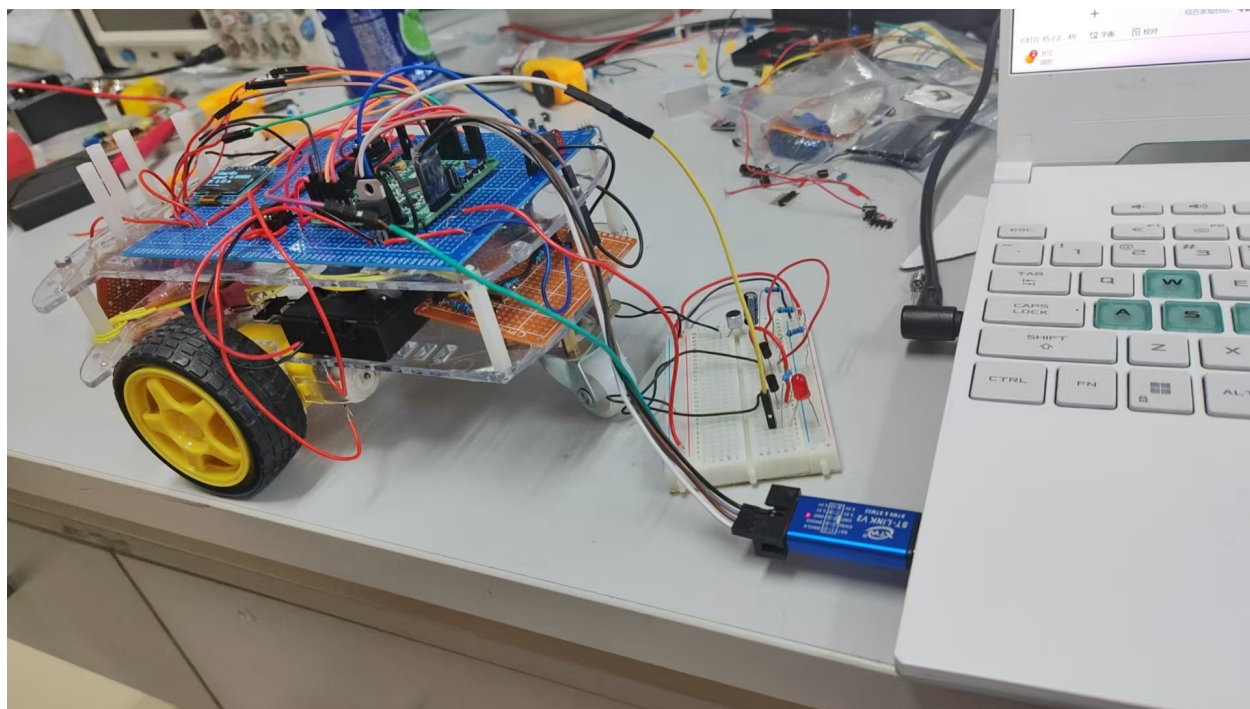


图 15: 整车焊接拼装效果图

5 总结

本设计成功实现了一套具备红外循迹、障碍物识别、计圈计时及状态显示功能的智能小车系统。**我们组完成了全部八项功能。**在调试过程中，项目团队多次遇到模块失灵、电压异常等问题，通过系统性排查予以解决，锻炼了嵌入式系统开发硬件电路分析等方面的综合能力。未来，若想继续深入研究此项目，根据四传感器设计更高级的寻迹算法，例如加权或者 PID 算法，是一些值得探讨的方向。

6 参考文献

1. CSDN 博客. H 桥驱动详解及代码实现. https://blog.csdn.net/qq_34802028/article/details/122431195
2. 立创商城. MX1919 9110H 双路直流电机驱动模块说明书. <https://item.szlcsc.com/19916709.html>
3. ST 中文社区. 红外循迹模块的调试与使用. <https://shequ.stmicroelectronics.cn/thread-634915-1-1.html>
4. CSDN 博客. 霍尔元件测速原理及 STM32 代码实现. https://blog.csdn.net/a_huan258147/article/details/107558419

5. 立创商城. LM339 比较器芯片技术手册. <https://item.szlcsc.com/517193.html>
6. 立创商城. 7805 三端稳压器规格书. <https://item.szlcsc.com/3426314.html>
7. 电工技术开发网. 光电门测速系统原理与设计. <http://www.dzkfw.com.cn/Article/MEMS/7147.html>
8. 博客园. 红外寻迹算法与状态机设计思路. <https://www.cnblogs.com/liangda/p/9901003.html>
9. CSDN 博客. 声控电路原理图与放大电路分析. <https://blog.csdn.net/oXiaoXue123456789/article/details/109903495>
10. CSDN 博客. 嵌入式 C 语言常用语法及陷阱总结. https://blog.csdn.net/best_xiaolong/article/details/108177668