

小车跟随行驶系统

摘要：本系统是以微控制器 STM32F103C8T6 为控制核心的小车跟随系统。小车通过 STM32F103C8T6 控制两轮差速驱动电机实现运动，利用基于树莓派的 OpenCV 视觉系统处理摄像头数据。距离控制采用在两车背后贴一红色方块，通过面积测量返回到 MCU 中调整小车速度的方法，实现对两车距离的控制。领头小车与跟随小车之间采用蓝牙模块进行通信并协同完成任务。串口屏作为显示和交互模块进行人机交互，实现用户对目标任务的选择功能。最终，本系统可实现领头小车和跟随小车的运动控制，跟随小车追踪领头小车且完全受领头小车控制，完成每项任务的行驶过程。

关键词：PID 算法；双车协同；人机交互

一、系统方案

1. 方案的选择和比较

1.1 电机选择方案

方案一：闭环步进电机。该方案的优点是步距稳定，响应速度较快，没有回程差，可以实时监测电机位置，控制精度高，存在位置反馈环节，能更好的跟踪输入信号的变化；缺点是成本较高，控制算法较为复杂，需要更长时间的调试与改善。

方案二：直流减速电机。该方案优点是转速范围广，具有较大的起动扭矩，传动比分级精细，体积小，寿命长，维护成本和制造成本相对较低，适用于大规模生产和应用；缺点是转速会受到电枢电压、负载和磨损等因素的影响，容易出现速度波动较大的情况。

方案比较与选择：本系统要求领头小车和跟随小车以不同速度巡线行驶，对转速调节范围要求较高，高速过弯时要求小车能快速调整两轮速度。综合考虑，直流减速电机转速范围宽泛，调速方式简单，在本题所设的场景中表现更为突出，因此选择方案二。

1.2 视觉处理平台选择方案

方案一：OpenMV。该方案优点是拥有丰富的示例代码，可以快速上手机器视觉应用，体积较为小巧，有利于携带和安装在各种设备上；缺点是硬件资源有限，无法实现某些复杂的机器视觉算法的需求，扩展性相对有限，缺乏成熟的第三方库和工具支持。

方案二：树莓派 5B。该方案的优点是处理速度快，运行内存大，便于处理复杂图像以及高精度的识别；缺点是启动电流较大，对降压模块要求较高，自启动困难，代码编写较为繁琐。

方案比较与选择：本题要求该系统能实时识别红色色块大小以实现保持车距的目的，对视觉处理的响应速度要求较高。综合考虑，树莓派 5B 搭载 OpenCV 视觉库在高精度视觉处理上更有优势，因此选择方案二。

1.3 显示模块选择方案

方案一：OLED 显示屏。OLED 显示屏的优点是对比度高，显示清晰，功耗和成本较低，体积小巧，便于安装；缺点是发光材料寿命相对较短，可能导致颜色失真和屏幕亮度不均，屏幕无法实现人机交互，且 IIC 通信对通信质量和稳定性要求较高。

方案二：串口屏。串口屏的优点是使用 UART 串行通信，显示清晰，功能丰富，支持触摸功能，更加方便人机交互；缺点是体积较大，占用空间，成本相对更高。

方案比较与选择：本题要求该系统具有人机交互功能，即领头小车需要实现用户自由选择目标任务的功能，同时串口屏相较于 OLED 可以显示更多的内容，功能更加强大，方便实现题目要求，因此选择方案二。

2. 小车跟随系统设计方案

系统以 STM32F103C8T6 为主控芯片，包含两个子系统，一个为领头小车，一个为跟随小车。该系统通过串口屏实现人机交互，用户可通过按键选择目标任务和启动小车，从而实现跟随小车追踪领头小车绕圈行驶的功能。领头小车上的串口屏识别到用户操作后将相应任务要求发送给树莓派和跟随小车的 MCU，树莓派通过摄像头识别地面上的黑色引导线和不同标志，将巡线偏移量、视野内红色物块的面积与目标面积大小差值及目标运动状态发送给 MCU；跟随小车的 MCU 收到指令后与领头小车保持一定距离，跟随领头小车行驶。当小车行驶在内圈时，点亮 LED 指示当前行驶状态；当小车完成一次行驶到达终点，领头小车和跟随小车通过蜂鸣器发出声音提示。整个小车跟随行驶的系统结构框图如图 1 所示。

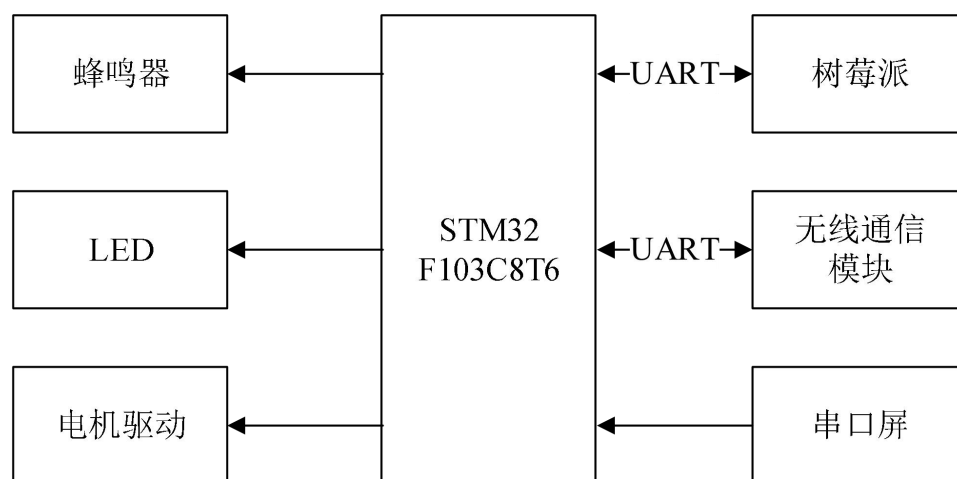


图 1 系统结构框图

二、理论分析和计算

1. 小车间通信模式分析

领头小车和跟随小车在完成任务时通过串口连接的 HC-05 蓝牙模块进行远程通信，通信内容包括当前任务编号、领头小车和跟随小车的位置情况、两车间距离以及当前任务速度设置等。串口波特率为 115200bps，通信格式为数据包头、数据类型、数据内容、数据校验位和数据包尾。通信频率为 20Hz。当领头小车和跟随小车的蓝牙模块都能够正常通信，即进入串口中断的频率达到 20Hz 时，领头小车和跟随小车才可以正常开始运行。

2. 小车运动控制设计

领头小车和跟随小车结构均为三轮前驱小车，其中一轮为万向轮，左右两轮为驱动轮。小车运动状态可以分为前后直行和顺逆旋转两种运动，若以前进和顺时针旋转为正方向，则小车直行速度和顺逆旋转角度可根据编码器反馈值计算得出，具体计算方式由式(1)和式(2)给出：

$$v_{straight} = \frac{v_{encoder1} + v_{encoder2}}{2} \quad (1)$$

$$\alpha_{turn} = \frac{S_{encoder1} - S_{encoder2}}{2d_{wheels}} \times 360^\circ \quad (2)$$

其中， $v_{straight}$ 是直行速度， α_{turn} 是顺逆旋转角度， $v_{encoder1}$ 、 $v_{encoder2}$ 、 $S_{encoder1}$ 和 $S_{encoder2}$ 是由编码器计算得出的左右轮速度和左右轮行驶距离， d_{wheels} 为两轮中轴线间距。

本系统使用并行 PID 控制小车巡线行驶。摄像头捕捉视野范围内的黑色引导线的轮廓，行驶过程中实时反馈黑色引导线的位置。以摄像头视野中黑色引导线位置与中心位置的偏差为巡线偏差环的输入量进行 PID 控制，输出结果为左右轮速度与实际目标速度之间的偏差量。速度环为基础环，以 $v_{encoder1}$ 左轮速度、 $v_{encoder2}$ 右轮速度与左右轮各自目标速度 v_{left_target} 和 v_{right_target} 的差值作为输入量进行 PID 控制，输出结果为 PWM 波的目标占空比。将左右轮的偏差量分别加在实际的目标速度 v_{target} 上即可得出左右轮各自的新目标速度，由速度环得到 PWM 波目标占空比，MCU 处理后输出给电机驱动模块，从而实现对小车以指定速度沿黑色引导线行驶的功能。具体过程由式(3)和式(4)给出：

$$v_{left_target} = v_{target} + v_{line_bias} \quad (3)$$

$$v_{right_target} = v_{target} - v_{line_bias} \quad (4)$$

3. 小车间距离控制

本系统使用并行 PID 控制小车间距离。前车后方粘贴一红色物块，初始时将两车间隔设置为 20cm，此时后车摄像头视野范围内的红色面积大小为标准面积值。行驶状态中后车摄像头实时反馈测量到的面积值，当实际面积值与标准面积值的差值为正时，表示两车距离过近，后车应减速；反之，表示两车距离过远，后车应加速。实时面积值与标准面积值的差值为间距控制环的输入量，该环的输出结果当前小车速度与目标速度之间的偏差量 v_{dis_bias} 。将该偏差量叠加到任务要求的目标速度 v_{set} 上得到运行过程中实际的目标速度 v_{target} ，实际的目标速度作为速度环的一个输入量，由速度环得出 PWM 波目标占空比，从而实现两车间的距离控制。具体过程由式(5)给出：

$$v_{target} = v_{set} + v_{dis_bias} \quad (5)$$

三、电路、程序与机械设计

1. 电路设计

电路使用 12V 锂电池供电，搭配 12V 转 5V 稳压模块。采用两路 BUCK 电路供电，使用三个船式开关分别控制电机电路，STM32 电路和视觉处理平台电路。具体电路结构如图 2 所示。

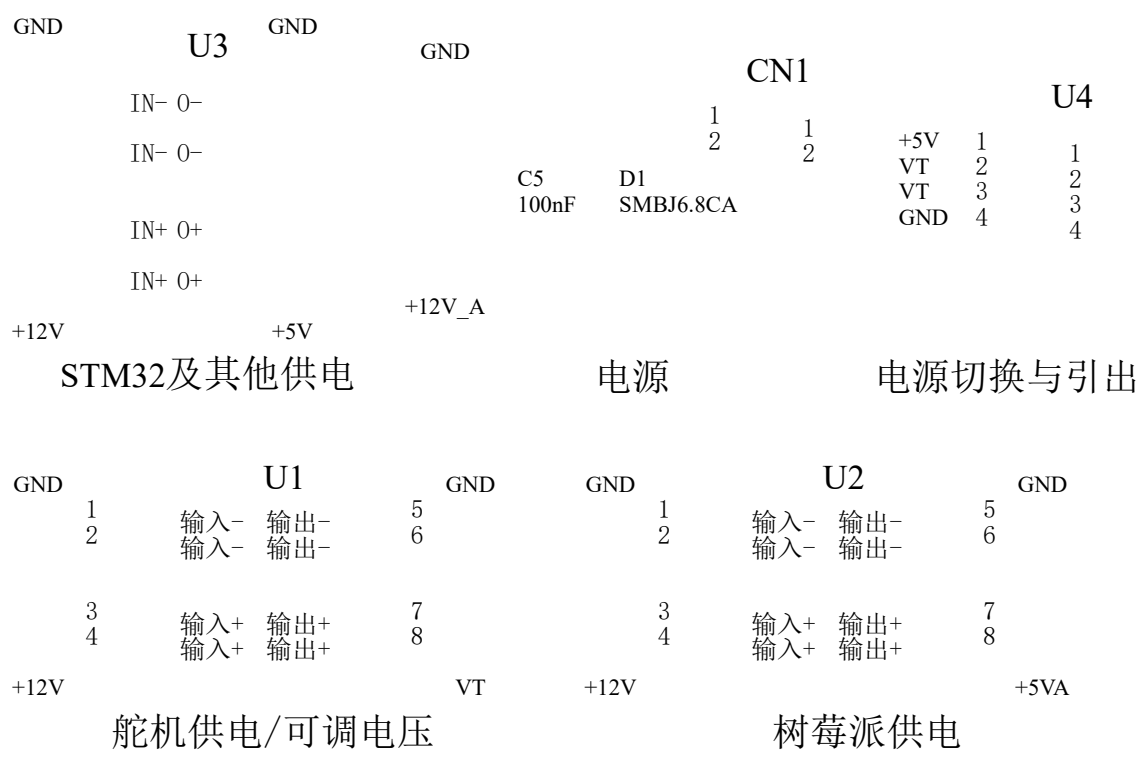


图 2 系统电路结构

2. 程序设计

2.1 程序整体框架

主体程序可分为两部分。一部分为领头小车的主体程序，另一部分为跟随小车的主体程序。

领头小车：用户在串口屏上选择目标任务，领头小车 MCU 解析串口屏发送的指令后将相应数据包传给领头小车树莓派和跟随小车 MCU，当检测到蓝牙模块可以正常通信后以指定速度沿线行驶。行驶途中领头小车树莓派实时反馈的巡线偏差值，领头小车 MCU 调整该偏差值左右轮电机转速。当领头小车完成一次行驶到达终点，发出声音提示；当领头小车行驶在内圈 BFD 段时，发出灯光指示。

跟随小车：跟随小车 MCU 收到领头小车指令后待命，当检测到蓝牙模块可以正常通信后以指定速度沿线行驶。行驶途中跟随小车树莓派实时反馈的巡线偏差值和领头小车车尾红色物块面积偏差值，跟随小车 MCU 调整该偏差值左右轮电机转速。当跟随小车完成一次行驶到达终点，发出声音提示；当跟随小车行驶在内圈 BFD 段时，发出灯光指示。

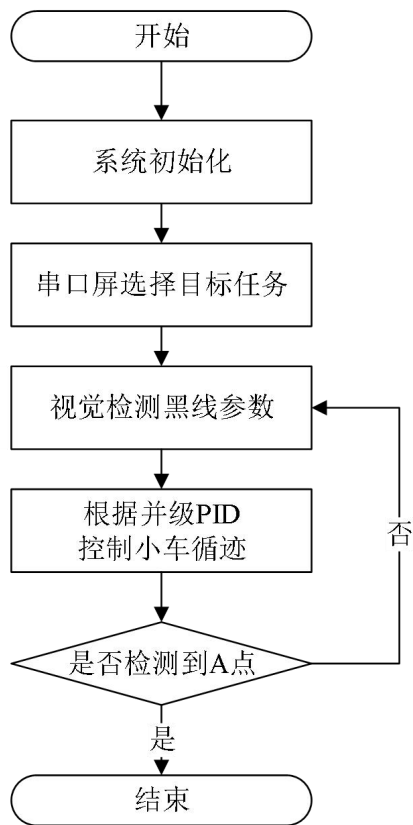


图 3 领头小车程序框图

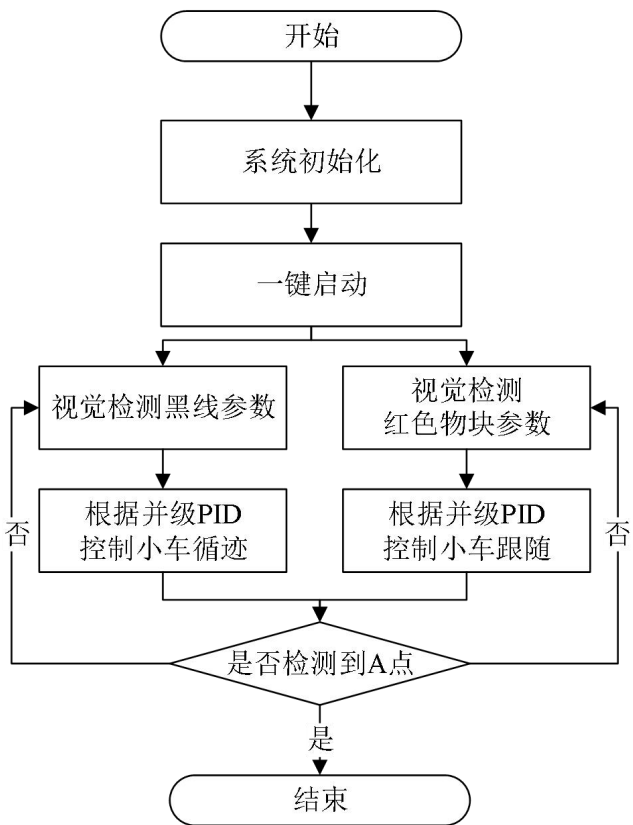


图 4 跟随小车程序框图

2.2 机器视觉程序设计

根据题目任务需要，本系统的视觉识别系统需要实现的功能主要是识别黑色引导线、识别红色物块面积和识别内外圈岔路口以及停车点。

识别黑色引导线先划分 ROI 区域，再使用阈值分割和开闭运算区分白色和黑色。识别黑色引导线轮廓后确定引导线中轴线的位置，以中轴线的位置作为黑色引导线的具体位置。

识别红色物块需先将图像转换为 HSV 颜色空间，再设置红色的 HSV 阈值范围，以便找到图像中红色物体的位置。然后使用阈值化后的图像，通过找到轮廓来定位红色物体的形状。最后计算目标物体轮廓在摄像头视野内的像素面积。

识别内外圈岔路口以及停车点首先划分 ROI 区域，设置黑色的 HSV 阈值范围。树莓派先对信号进行平滑化处理或使用小波变换来寻找峰值，接着利用 `scipy.signal.find_peaks()` 函数检测信号序列峰值，区分黑色引导线、内外圈岔路口和各停车点标志。内外圈岔路口表现为图片横向出现两个黑色峰值；停车点表现为横向上连续出现一段大于引导线宽度的黑色峰值。

3. 机械设计

该系统的机械结构如下图所示：

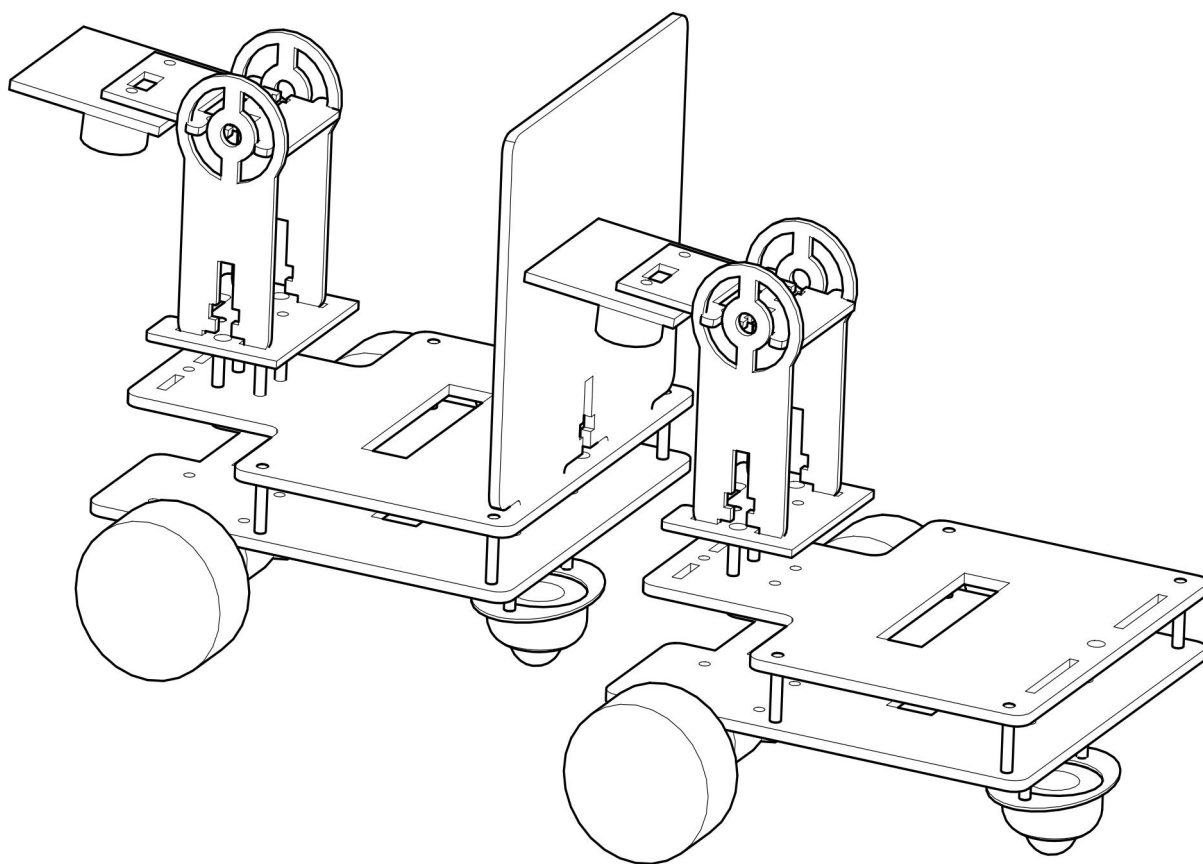


图 5 小车跟随系统机械图

小车使用加工好的环氧板和铜柱搭建支撑结构，使用普通车轮作为驱动轮，球头万向轮作为辅助从动轮，采用前驱结构以使小车可以更加稳定向前行驶。领头小车后方固定一块环氧树脂板，并在其上粘贴一红色方块，用于跟随小车判断二者距离，摄像头通过肋板固定在车前方，可调节摄像头角度，并且可以通过调整下方铜柱长短调节摄像头高度，使地图黑线与前车红色物块完整地出现在摄像头视野中，用于巡线与固定距离的跟随任务。

四、测试方案与测试结果

1. 测试环境

表 1 测试器件表

类型	品牌	型号
卷尺	得力	8201 卷尺
秒表	卡西欧	HS-70W 型秒表

2. 测试方案与结果

2.1 将领头小车放在路径的起始位置 A 点，跟随小车放在其后 20cm 处，设定领头小车速度为 0.3m/s，沿着外圈路径行驶一圈停止，测量两小车行驶所需时间与停止时两车距离。测量结果如表 2 所示。

表 2 小车跟随系统任务 1 测试表

次数	领头小车 行驶时间	跟随小车 行驶时间	领头小车 平均速度	两车距离	是否碰撞
1	20.56s	20.74s	0.292m/s	20.6cm	否
2	20.08s	20.31s	0.299m/s	18.9cm	否
3	19.69s	20.12s	0.305m/s	21.2cm	否

2.2 将领头小车放在路径轨迹的起始位置 A 点，跟随小车放在路径上 E 点所在边的直线区域，由测试专家指定的位置，设定领头小车速度为 0.5m/s，沿着外圈路径行驶两圈停止，测量两小车行驶所需时间与停止时两车距离。测量结果如表 3 所示。

表 3 小车跟随系统任务 2 测试表

次数	领头小车 行驶时间	跟随小车 行驶时间	领头小车 平均速度	两车距离	是否碰撞
1	24.46s	24.68s	0.491m/s	21.3cm	否
2	23.94s	24.22s	0.501m/s	19.1cm	否
3	24.15s	24.41s	0.497m/s	20.6cm	否

2.3 将领头小车放在路径的起始位置 A 点，跟随小车放在其后 20cm 处，领头小车和跟随小车连续完成三圈路径的行驶。第一圈领头小车和跟随小车都沿着外圈路径行驶。第二圈领头小车沿着外圈路径行驶，跟随小车沿着内圈路径行驶，实现超车领跑。第三圈跟随小车沿着外圈路径行驶，领头小车沿着内圈路径行驶，实现反超和再次领跑。测量两小车行驶所需时间与停止时两车距离。测量结果如表 4 所示。

表 4 小车跟随系统任务 3 测试表

次数	领头小车 行驶时间	跟随小车 行驶时间	两车距离	是否碰撞	是否顺利 两次超车
1	29.32s	29.58s	20.3cm	否	是
2	28.64s	28.82s	20.2cm	否	是
3	28.76s	28.97s	19.7cm	否	是

2.4 由测试专家在路径的 E 点所在边的直线区域指定位置，放上“等停指示”标识。然后，将领头小车放在路径的起始位置 A 点，跟随小车放在其后 20cm 处，设定领头小车速度为 1m/s，沿着外圈路径行驶一圈，行驶中两小车不得发生碰闯。

表 5 小车跟随系统任务 4 测试表

次数	领头小车 行驶时间	跟随小车 行驶时间	停车距离	停车时间	是否碰撞
1	11.07s	11.29s	1.3cm	5.26s	否
2	11.56s	11.72s	1.2cm	5.13s	否
3	11.23s	11.44s	0.7cm	5.09s	否

3. 测试结果分析

经过多次测试与调试，小车跟随系统可以较好地达成任务要求以及预期效果，整体大小符合要求，在测试中均能实现双车保持一定距离跟随，两车通信没有问题，在行驶中未发生碰撞，可以完成不同速度的行驶，也可完成行驶内外轨道的切换以及准确停止，运动速度稳定，精度较高，两车停车时间差较小，同时也有一定对反光的抗干扰能力。