

南京大學

智能小车设计报告

姓 名 邓 世 豪

学 号 191220021

所在院系 计算机科学与技术系

年 级 2019 级

指导老师 方 元

邮 箱 191220021@smail.nju.edu.cn

2022 年 1 月 12 日

摘要：随着智能系统和计算机的飞速发展，当今社会已经越来越离不开“智能”产品。本次报告中的智能小车（型号为 **Waveshare AlphaBot2**）是基于树莓派（**Raspberry Pi**）开发的，智能小车上集成了红外传感器、超声波测距器、蜂鸣器、操纵杆、直流电机（用于控制小车轮子）、彩灯等部件。智能小车以树莓派作为控制核心，各种传感器组成环境感知系统，最终实现了红外遥控小车行驶，红外避障自动行走，红外循迹自动行走等功能。整个系统相当于一个简单的智能系统，能够感知特定环境并做出系统认为最优化的响应。

关键词：智能系统嵌入式应用，树莓派，红外，避障，循迹，Python

目录

1 方案设计	4
1.1 总体设计理念	4
1.2 硬件的选择和设计	4
1.3 程序的设计	4
2 智能小车介绍	5
2.1 硬件介绍	5
2.1.1 蜂鸣器	5
2.1.2 NEC 红外遥控器	5
2.1.3 直流电机	5
2.1.4 红外传感器	6
2.1.5 彩灯	6
2.2 程序介绍	7
2.2.1 代码设计总览	7
2.2.2 GPIO 模块和 PWM 模块	7
2.2.3 Buzzer 模块	7
2.2.4 Color 模块和 LED 模块	8
2.2.5 WheelControl 模块	8
2.2.6 irsensors 模块	8
2.2.7 ircontrolmove 模块	9
2.2.8 linetracking 模块	10
2.2.9 iravoidance 模块	10
3 总结与反思	11
4 参考	11
5 附录	11
5.1 程序代码	11
5.2 视频展示	11

1 方案设计

1.1 总体设计理念

将整个智能小车系统分为输入（感知）、运算（处理）、输出（反馈）三个部分。智能小车的硬件主要负责感知和反馈这两个部分，程序（软件）主要负责系统的运算处理。这样的设计有利于硬件和软件的搭配分工，同时能将各自的特点发挥出来。

1.2 硬件的选择和设计

设计理念：硬件在整个智能系统中作为输入和输出，只负责对特定环境的感知和反馈，不参与具体的控制以及运算。

智能小车上的传感器主要作为整个系统的输入（反馈），包括小车底部的 5 个反射式红外光电传感器，小车前方的 2 个反射式红外光电传感器以及小车顶层的 1 个红外接收头。其中车底的 5 颗红外传感器用于自动循迹，小车前方的 2 颗红外传感器用于避障行驶，小车顶层的 1 个红外接收头用于接收 NEC 红外遥控器发射的信号。

智能小车上的控制器主要作为整个系统的输出（反馈），包括小车底部的 4 颗全彩 RGB LED 灯珠，小车前方的 2 颗避障信号灯，小车底部的 2 个橡胶轮，小车顶层的蜂鸣器。其中车底的 4 颗 LED 灯珠用于炫彩的灯光输出，小车前方的 2 颗避障信号灯作为标识配合红外避障行驶功能，小车底部的 2 个橡胶轮配合小车底部的 2 颗万向轮在小车的 N20 微型减速直流电机的控制下完成小车的行驶功能，小车顶层的蜂鸣器用于特殊情况的警示输出。

1.3 程序的设计

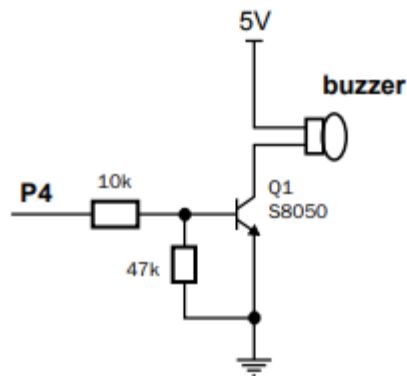
设计理念：程序（软件）是智能小车的核心控制部分，主要负责系统的运算（处理）任务。总体程序代码采用了代码模块化和部分面向对象的设计理念，以 Python 作为基本语言，不同的硬件对应设计了的控制代码以便做好各个功能的分离，以期能实现代码的模块化，各模块内高内聚，各模块之间低耦合，同时也为多线程编程奠定基础。具体的设计细节将在后面详细介绍。

2 智能小车介绍

2.1 硬件介绍

2.1.1 蜂鸣器

蜂鸣器是一种一体化结构的电子发声设备，主要有压电式和电磁式两种类型。



多谐振荡器通电源后输出 1.5~2.5kHz 的音频信号，推动压电蜂鸣片发声。

图 1 蜂鸣器

2.1.2 NEC 红外遥控器

红外遥控器发射的信号使用 38kHz 左右的载波对基带进行调制。接收端对信号监测、放大、滤波、解调等一系列处理，然后输出基带信号。收发方采用约定的协议进行通信。发送端首先发送一个 9ms 低电平接 4.5ms 高电平的引导码，接收方检测到引导码后开始识别后面的数据：0.56ms 低+0.56ms 高表示“0”，0.56ms 低+1.69ms 高表示“1”。一组 0、1 序列构成一个按键特征字。接收方根据收到的特征字产生一定的动作。

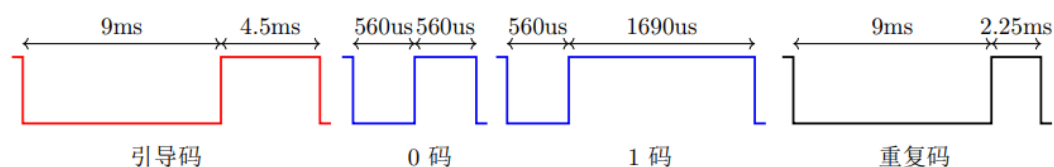


图 2 NEC 红外遥控器协议

2.1.3 直流电机

直流电机控制电路由两个功率输出端和一个控制开关组成。当控制开关接通时，输出电流方向由两个输入端的电压差决定，从而实现电机的双向转动控制。

调整开关通、断时间比例即可调整电机转速。

该智能小车的左右轮的控制电机分别由 GPIO12, GPIO13 和 GPIO21, GPIO23 作为输出端, 分别由 GPIO6 和 GPIO26 作为控制端。用 PWM 模块的方法来控制控制端, 就可以实现电机转速的调整从而控制智能小车的橡胶轮转速(将在程序介绍中详细说明)。

2.1.4 红外传感器

红外传感器由一个红外发射管和一个红外接收管组成。发射管和接收管之间有一块黑色挡板隔开形成光隔离防止互相干扰。接收管可以接收发射管发射的红外光经过反射后的红外光, 并根据反射的红外光的强度做进一步的处理。接收强度与传播距离和反射面对红外光的吸收性质有关。

在循迹功能中。发射管发出的红外光被反射后由接收管接收, 再经过模数转换器得到量化的红外强度值。具体的数值可以作为判断智能小车是否偏离黑色轨迹的依据。

在避障功能中, 发射管发出的红外光被反射后由接收管接收, 并根据红外强度直接从 GPIO16 (左侧红外) 和 GPIO19 (右侧红外) 输出 “0” 或 “1”。另外, 可以通过旋转调整小车底部的两个电位器来调节红外避障的感应距离。

2.1.5 彩灯

每一颗彩灯灯珠可以通过红、绿、蓝三个发光二极管产生的不同亮度并形成不同的颜色。每个发光二极管由一个 8 位的数字信号控制, 并实现 256 级亮度。

在小车彩灯的控制上利用了 rpi_wa281x 模块, 它同时可以使用 PWM, 并提供了 Python 的接口, 非常方便使用。

2.2 程序介绍

2.2.1 代码设计总览

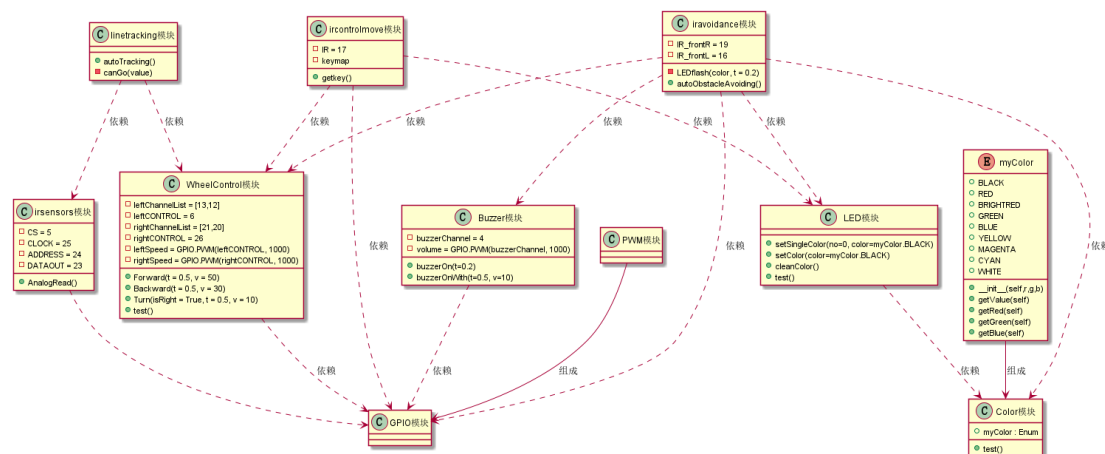


图 3 程序代码模块图

图 3 是用 UML 语言建模的程序代码图，它粗略地描述了模块之间的关系并展现了各个模块的大体内容。图中，实线箭头标识了部分与整体的关系，虚线箭头标识了模块之间的依赖关系。另外，模块中定义的 `test()` 方法是作为 `main` 程序时用来测试的方法，没有其他特殊的功能。下面将一一介绍各模块的设计思路和实现方法。

2.2.2 GPIO 模块和 PWM 模块

RPi.GPIO Python 模块是 Ben Croston 编写的用于控制树莓派 GPIO 接口的函数库，同时也是官方发布的树莓派 Linux 系统默认安装的模块。有了这个模块就能非常方便的控制树莓派上的各个 GPIO 引脚。PWM 是 GPIO 模块中非常实用的模块，它通过设置频率（单位：Hz）和占空比（0 到 100）可以控制某个 GPIO 引脚实现 PWM 功能。

2.2.3 Buzzer 模块

Buzzer 模块依赖于 GPIO 模块，通过设置特定 GPIO 引脚（由变量 `buzzerChannel` 持有）的输出电平，控制硬件蜂鸣器的开关。另外还可以通过设置 PWM 功能来调整蜂鸣器的音量（由变量 `volume` 持有），但实际效果不是很理想，可能是由于控制蜂鸣器的 4 号引脚不支持硬件 PWM 而是通过软件安排线程实现，因而稳定性和准确性都没法保证。

2.2.4 Color 模块和 LED 模块

Color 模块主要设计来支持 LED 灯珠颜色的设置。在 Color 模块中定义了枚举类型 `myColor`，其中包含了常用的 9 种颜色，并支持设置 R、G、B 来自定义新的颜色，方法 `getValue()` 返回 color 的 16 位 RGB 数值用来和 LED 模块中的接口衔接起来。

LED 模块包含了 `neopixel` 模块中的函数库，同时也依赖于 Color 模块。前者可以通过 RGB 值设置每颗灯珠的颜色，同时可以通过 PWM 功能设置灯珠的亮度。另外，LED 模块中定义封装了 `setColor()`、`cleanColor()` 等方法用来更方便地设置智能小车的 4 颗彩灯。

2.2.5 WheelControl 模块

WheelControl 模块依赖于 `RPi.GPIO` 模块，通过设置树莓派的 GPIO 引脚来控制直流电机从而控制智能小车的轮子。其中，通过设置直流电机控制端对应的 GPIO 接口的 PWM 功能，可以调节电机转速。`leftChannelList` 和 `rightChannelList` 两个列表指明了控制左轮和右轮的直流电机的两个电压功率输出端（0 或 1），`leftControl` 和 `rightControl` 两个变量指明了控制左轮和右轮的直流电机的控制端。

该模块中定义的 `Forward()`、`Backward()`、`Turn()` 三个方法分别可以控制智能小车前进、后退和转向。其中，方法的参数 `v`（0 到 100）表示车轮速度，它是通过调节 PWM 中的占空比参数来实现的；参数 `t`（单位：s）表示车轮转动的持续时间；`Turn()` 中的参数 `isRight` 则表示小车的转向（默认右转）。

2.2.6 irsensors 模块

`irsensors` 模块依赖于 `RPi.GPIO` 模块，用软件仿真 SPI (Serial Port Interface) 协议：在 `IOCLK` 产生时钟脉冲，`ADDR` 逐位输入通道编码，从 `DOUT` 读出转换值。每次读取采样的 10bit 数据操作中，前 4bit 写入待采样的通道编码。由于通道编码随本轮采样过程中输入，因此每次采样通道的数据实际上对应着上一轮的通道编码，采集六个通道的数据，第一个通道是无效的。

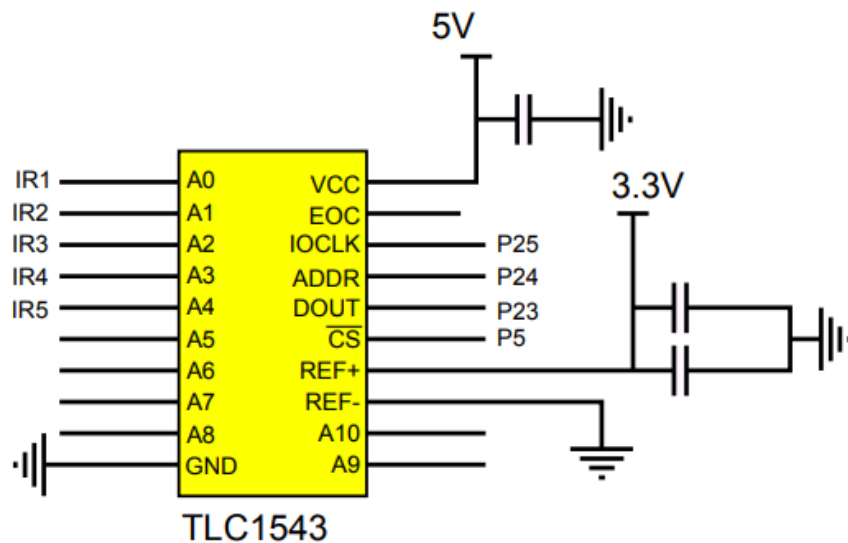


图 4 红外数模转换器

在该模块中，CS、CLOCK、ADDRESS、DATAOUT 这 4 个变量标识了对应的 GPIO 引脚，方法 `AnalogRead()` 返回一个包含 5 个元素的列表，这 5 个数值表明了对应的小车底部 5 个红外接收器所接收到的红外强度。

2.2.7 ircontrolmove 模块

ircontrolmove 模块实现了红外遥控器控制智能小车进行移动、亮灯、鸣笛等功能。

为了实现上述功能，该模块依赖于 GPIO 模块、LED 模块、Buzzer 模块、WheelControl 模块。根据 NEC 红外遥控器协议，本模块中 IR 变量持有硬件红外接收头对应的 GPIO 引脚用来接收遥控信号，keymap 字典型变量则保存了对应的遥控器按键与数值的键值对，方法 `getkey()` 返回接收到红外信号对应的键值。在方法 `getkey()` 中，采用了 GPIO 的高级输入功能，以阻塞方式等待 IR 引脚上升沿的出现，以此来标识一个红外信号的开始接收。

为了能够同时启动智能小车的 LED 灯珠、蜂鸣器以及直流电机，该模块运用了多线程的方法。首先通过 `getkey()` 方法拿到输入，不同的输入对应不同的功能。例如，键值 0x16（即“0”键）对应开启 LED 彩灯展示，具体的实现为创建一个新的线程去调用 LED 模块中的 `myLED.test()` 方法，此前通过对 LED 线程对象的判断来防止上一个 LED 线程还没跑完就又重新开启一个 LED 线程引起的 Race condition。类似的，蜂鸣器的功能和小车移动的功能都是通过创建新的线程调用对应方法来实现的（[参考具体代码](#)）。

2.2.8 linetracking 模块

linetracking 模块实现了智能小车自动循迹的功能。

循迹功能基于小车底部的 5 个红外传感器，因此 linetracking 模块依赖于 irsensors 模块，通过为了能控制小车移动，linetracking 模块还依赖于 WheelControl 模块。

由于黑色材质对红外光的吸收较多，因此反射的红外信号强度弱，对应返回的数值小，智能小车就是根据这个原理来判断是否处在轨迹上的。

本模块中，方法 canGo() 返回一个 bool 型变量，依据 irsensors 模块中的 AnalogRead 方法返回的数值来判断表明当前位置是否可以继续前行。方法 autoTracking() 主体为一个死循环，首先通过 canGo() 判断当前位置是否可以前行，若能则调用 WheelControl 模块中的 Forward 方法前行一小段距离并进入下一次循环；若不能，则根据 AnalogRead 方法返回的数值来判断应该左转还是右转并调用相应的方法然后进入下一次循环（[参考具体代码](#)）。

[戳我看视频展示](#)

2.2.9 iravoidance 模块

iravoidance 模块实现了智能小车自动避障行驶的功能。

避障功能主要依靠小车前方的两个红外传感器，在靠近物体时，接收器接收到反射的足够强的红外光，对应 GPIO 引脚输出“0”，相反输出“1”。通过旋转调整小车底部的两个电位器可以调节恰当的感应距离。根据 GPIO 引脚的输出，对小车做出相应的控制来进行实时反馈。

本模块中，IR_frontR、IR_frontL 变量分别标识出红外传感器对应的 GPIO 引脚，LEDflash() 方法则对 LED 模块中的方法进一步封装，有助于实现小车不同状态闪烁不同灯光的功能。在 autoObstacleAvoiding() 方法中，首先通过 GPIO.input 获得红外接收器的输入，根据两个的输入（两侧红外接收器），做出前进、左转、右转、旋转半周的行动。

为了能使智能小车在移动的过程中同时闪烁 LED 灯珠，在危险的情况下能同时响起蜂鸣器，本模块同样使用了多线程编程的方式（[参考具体代码](#)）。

[戳我看视频展示](#)

3 总结与反思

本次智能小车的设计与完成实现了主要的基本功能，包括 LED 灯珠的控制、蜂鸣器的控制、小车的移动功能、小车的自动循迹功能、小车的遥控实现、小车的自动避障功能。同时在整个设计中，偏向于程序代码的设计与实现，做到了代码的模块化，模块之间低耦合，模块内高内聚，同时代码的可扩展性较强，并且实现了基本的并发编程。最终也呈现出一个算是智能的系统。

但本次设计也有很多不足和遗憾。比如循迹功能中，在不同的地面上行走需要重新调节脚本的参数，代码还不够完善。此外，没有使用上超声波测距器、舵机、摄像头等硬件，没来得及完成图形化控制软件的开发是本次实验的诸多遗憾。

4 参考

[1]方元.Python[CP].南京大学电子科学与工程学院, 2021.2

[2]廖雪峰.Python 教程[OL].<https://www.liaoxuefeng.com/wiki/1016959663602400>

5 附录

5.1 程序代码

所有代码均上传至 [Github 代码托管网站个人主页](#)。

5.2 视频展示

视频展示均上传至 [Bilibili 视频网站个人主页](#)。

- [循迹展示视频](#)
- [避障展示视频](#)